Journal of Systems Engineering and Electronics
Vol. 28, No. 2, April 2017, pp.374 – 384

Tag clustering algorithm LMMSK: improved K-means algorithm based on latent semantic analysis

Jing Yang and Jun Wang*

School of Economics and Management, Beihang University, Beijing 100191, China

Abstract: With the wide application of Web_2.0 and social software, there are more and more tag-related studies and applications. Because of the randomness and the personalization in users' tagging, tag research continues to encounter data space and semantics obstacles. With the min-max similarity (MMS) to establish the initial centroids, the traditional K-means clustering algorithm is firstly improved to the MMSK-means clustering algorithm, the superiority of which has been tested; based on MMSK-means and combined with latent semantic analysis (LSA), here secondly emerges a new tag clustering algorithm, LMMSK. Finally, three algorithms for tag clustering, MMSK-means, tag clustering based on LSA (LSA-based algorithm) and LMMSK, have been run on Matlab, using a real tag-resource dataset obtained from the Delicious Social Bookmarking System from 2004 to 2009. LMMSK's clustering result turns out to be the most effective and the most accurate. Thus, a better tag-clustering algorithm is found for greater application of social tags in personalized search, topic identification or knowledge community discovery. In addition, for a better comparison of the clustering results, the clustering corresponding results matrix (CCR matrix) is proposed, which is promisingly expected to be an effective tool to capture the evolutions of the social tagging system.

Keywords: tag clustering algorithm, K-means, latent semantic analysis (LSA), min-max similarity (MMS), social tagging.

DOI: 10.21629/JSEE.2017.02.18

1. Introduction

Currently with the wide application of Web_2.0 and social software, the organization and management of network resources gradually consider collective wisdom [1 – 3]. In the second generation of the Internet, as represented by Twitter, CSDN, Delicious, YouTube, Flickr and Technorati, tagging is a process, in which users use the words that they prefer or the terms that they are familiar with as personalized tags for particular resources, according to their own understanding [4]. Therefore social tagging (also noted as collaborative tagging or folksonomy) has become the most important organizational resource in the Web_2.0 era [5 – 7].

In folksonomy, a tag is a set of free text keywords [8]. In fact, each labeled resource object is a document that contains certain valuable information, and the users' annotations of these documents reflect their interests or understandings regarding these documents [9,10]. By analyzing and mining the deep information of tag data, researchers can obtain the tag users' preferences or interests and the label resource objects' content, and can reveal the resource objects' theme to study the potential social relation network [11 – 13]. By taking advantage of the tag information, researchers can even realize personalized tag recommendation for users [14,15].

Although social tags are very useful, there are several obstacles that may hinder the applications of social tags in knowledge sharing and intelligent service, due to the free-form nature of tagging and the lack of explicit semantics in social tagging systems: syntactic variations, polysemy, synonym, and hypernym/hyponym [16]. To improve research, researchers first consider tag clustering as the means to solve these problems. Currently, there are many clustering methods that can be used in tag research: traditional K-means and improved K-means [17 – 21], hierarchical clustering and its improvements [22 – 24], and the clustering algorithms based on latent semantic analysis (LSA) [25 – 28], probabilistic LSA (PLSA) [27,28] and term frequency/inverse document frequency (TF-IDF) [25,29].

Not considering the semantic relations between tags, the results of traditional clustering algorithms, such as K-means and their improvements, miss the real but latent semantic association. However, compared with statistical methods, other algorithms based on LSA or PLSA lack accuracy. This paper presents a new tag clustering

Manuscript received January 05, 2016.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (71271018; 71531001).

method, LMMSK, and the comparison experiments prove that LMMSK outperforms k-means, MMSK-means, and LSA.

The overriding contributions of this paper are as follows:

(i) Preliminarily improve the traditional K-means clustering algorithm by choosing initial centroids with an MMS. The improved algorithm, called MMSK-means, provides a more accurate clustering result, and it is more stable since the results of multiple repeated runs are consistent.

(ii) The optimized tag clustering algorithm, LMMSK, retains the merits of both LSA and MMSK-means, since it is stable and accurate as with MMSK, and has practical significance for considering the semantic connection as with LSA.

(iii) For a better comparison of the clustering results, this paper proposes the clustering corresponding results matrix (CCR matrix). Through the CCR matrix, the authors can analyze the accuracy of clustering results and make comparisons between the results of two different algorithms.

The remainder of this paper is organized as follows. Section 2 provides the state of related work. Section 3 discusses improvements in theoretically choosing the initial centroids in K-means theoretically. Section 4 combines the improved K-means algorithm with LSA and proposes a new tag clustering method. Section 5 discusses and compares the superiority of the new tag clustering method. Finally, Section 6 provides the conclusion and perspectives for future research.

2. Related work

There are many studies related to social tags, including researches and applications of the social tagging system, users and resources of the system, and the relation between tags, users and resources. Since the main research content is the tag clustering algorithm, this paper provides the statement of research on tags' application and clustering methods as related works.

2.1 Social tag application

Since social tag is one of the most important elements in folksonomy, researchers have conducted a number of studies on the use and applications of social tags. Gemmell et al. proposed a type of unsupervised hierarchical clustering method based on the user's interest model and applied this method in personalized search [30]. Using a K-means clustering algorithm, Hayes et al. identified the most topic-relevant blogs in the cluster by using social tags [31].

In social tagging systems, social tags can be used as research object to study the knowledge community. There are many studies on the knowledge community, including static researches, such as the studies on community structure by Newman et al. [32,33] and Barabási et al. [34,35], and dynamic research, such as studies on the community structure evolution [36 – 38]. The research in this field has gradually matured, whereas few studies from social tag perspective exist. Using latent information in social tags can provide a new means for researchers in the community study.

2.2 Social tag clustering

The application of tags includes topic identification, personalized search/recommendation, and community discovery and evolution. However, efficient tag clustering is needed before application. There are many methods to achieve tag clustering, including traditional methods, such as K-means [17 – 19] and the hierarchical clustering algorithm [22 – 24], and their improvements or extended algorithms, such as ant K-means (AK), which was proposed to optimize the resulting set of non-overlapping clusters based on the total within cluster variance (TWCV) [20,21].

Clustering algorithms based on text data processing, are mainly used in information retrieval and text classification [25]. A text similarity clustering algorithm based on TF-IDF attains relevance among terms, documents and particular classifications by assessing the importance of the words in the text [25,29]. LSA was first proposed by Deerwester et al., and the researchers utilized the implicit higher-order structure in the association of terms with documents ("semantic structure") to improve the detection of relevant documents based on terms found in queries by using singular value decomposition (SVD) [26]. In [25], the authors conducted an experiment to compare three different types of text classification methods: TF-IDF, latent semantic index (LSI) (namely, LSA) and multi-words, and LSI performed best. Based on LSA, Hoffman presented a new algorithm, PLSI (namely, PLSA), by introducing probability as a new variable in statistics' perspective [27]. Using PLSA to improve a collaborative filtering recommendation, Popescul et al. analyzed the resource-tag relation and the user-tag relation simultaneously [28]. In 2006, Begelman et al. published their idea that there is a strong correlation between co-occurrence tags, and they proposed a social tag clustering method based on tag co-occurrence [39]. Additionally, Giannakidou et al. completed social tag clustering after performing data analysis to glean a semantic correlation among the co-occurrence of users, tags and resources [40].

In addition, certain scholars attempted to combine different clustering algorithms to obtain a new algorithm, and

combined the advantages of two or more algorithms, such that the clustering effect is better than any single method. The authors preffered a hybrid algorithm to reduce the dependence of the initial centroids in K-means by combining K-means and harmony search clustering, which globally optimizes the clustering results rather than locally optimizing them [41]. Nguyen et al. subtly united the best characteristics of the K-means and expectation maximization (EM) algorithm, and proposed a clustering algorithm called genetic algorithm K-means logarithmic regression expectation maximization (GAKREM), which avoids their weaknesses, such as the need to specify initial number of clusters, termination in local optima, and lengthy computations [42].

In contrast to the hybrid algorithms noted above, this paper uses a social tag clustering perspective based on K-means and LSA to provide an innovative method. Utilizing the MMS to improve the choice of initial centroids in K-means, the authors acquire a new method called MMSK-means to achieve global optimization and make the results more robust. The authors then determine k, the cluster number, in K-means by combining MMSK-means with LSA, and thus, an innovative method is created.

3. K-means and its preliminary improvement

3.1 Traditional K-means

The K-means clustering algorithm is one of the ten classical algorithms in the data mining field, and, as a traditional clustering algorithm, similar to hierarchical clustering, it remains widely used. The basic idea of the K-means algorithm is relatively simple. The idea is summarized as (i) assignment, choose k initial centroids, count the distances between every data point and every centroid, and assign every data point to the nearest centroid; (ii) clustering, all of the data points assigned to the same centroid consist of a cluster; (iii) updating, update every centroid using the data points assigned to the centroid; and (iv) reiteration, reiterate the assignment of data points and the updating of centroids until minimal change in the clusters or centroids exists.

K-means is a widely used clustering algorithm, which shows its certain advantages in clustering. However, particularly in choosing of the initial centroids, K-means also has certain shortcomings.

The results of clustering are always expected to be better, the larger the difference among the different clusters, the smaller the difference in the same cluster. In Euclidean space, this difference is assessed by sum of the squared error (SSE), which counts the Euclidean distance between every data point and its nearest centroid and adds all of these distances to the sum of the squared error. In other words, after repeatedly running K-means many times, the cluster result with the minimum SSE is selected. The SSE in Euclidean space is estimated as follows:

$$\mathrm{SSE} = \sum_{i=1}^{k} \sum_{x \in C_i} dist(x, c_i)^2 \tag{1}$$

where c_i is the centroid of the ith cluster. By reiterating the assignment of data points and the updating of centroids, K-means continuously optimizes SSE. However, it is likely simply a local optimal because the optimization is based on chosen centroids rather than on all possible situations.

In document data processing, K-means is also usually used for clustering. In contrast to the Euclidean space, clustering is based on the cosine similarity and, corresponding to SSE in document data processing, the total cohesion is a measure of cluster results, which is counted as follows:

$$\mathrm{TotalCohesion} = \sum_{i=1}^{k} \sum_{x \in C_i} \cos(x, c_i). \tag{2}$$

Similarly, the optimal result of total cohesion, which is reached by reiterating the assignment of data points and the updating of centroids, most likely contains only a local optimal result.

The behavior of the K-means algorithm is mostly influenced by the number of specified clusters and the random choice of initial cluster centers. There are two problems in K-means, one is how to set the value k, and another is how to choose the k initial centroids. How many clusters do we need to make the result better? Additionally, how can we make the choice right for global optimization? In the next section, this paper will preliminarily improve K-means to answer the second question. And then, by introducing LSA, we will answer the first question.

3.2 MMSK-means: improved K-means

Postulating that the value of k is fixed, there are many means by which to choose the k initial centroids, such as randomly selecting initial centroids and opting for clusters with a minimum SSE or a maximum total cohesion. Randomly selecting initial centroids may slightly ameliorate the cluster results, however, it cannot guarantee global optimization. In addition, this process costs more time and space. Our improvement in K-means is fit to the similarity, considering that our object is document data. The improved algorithm, MMSK-means, is for k best initial centroids. Additionally, Fig. 1 provides the steps of the improved portion of the MMSK-means, whereas the remaining steps are the same using K-means. For better illumination, let kn be the number of centroids obtained in the process, and $\boldsymbol{C}$ is an $M * N$ matrix, which contains M points represented in N dimension vectors.

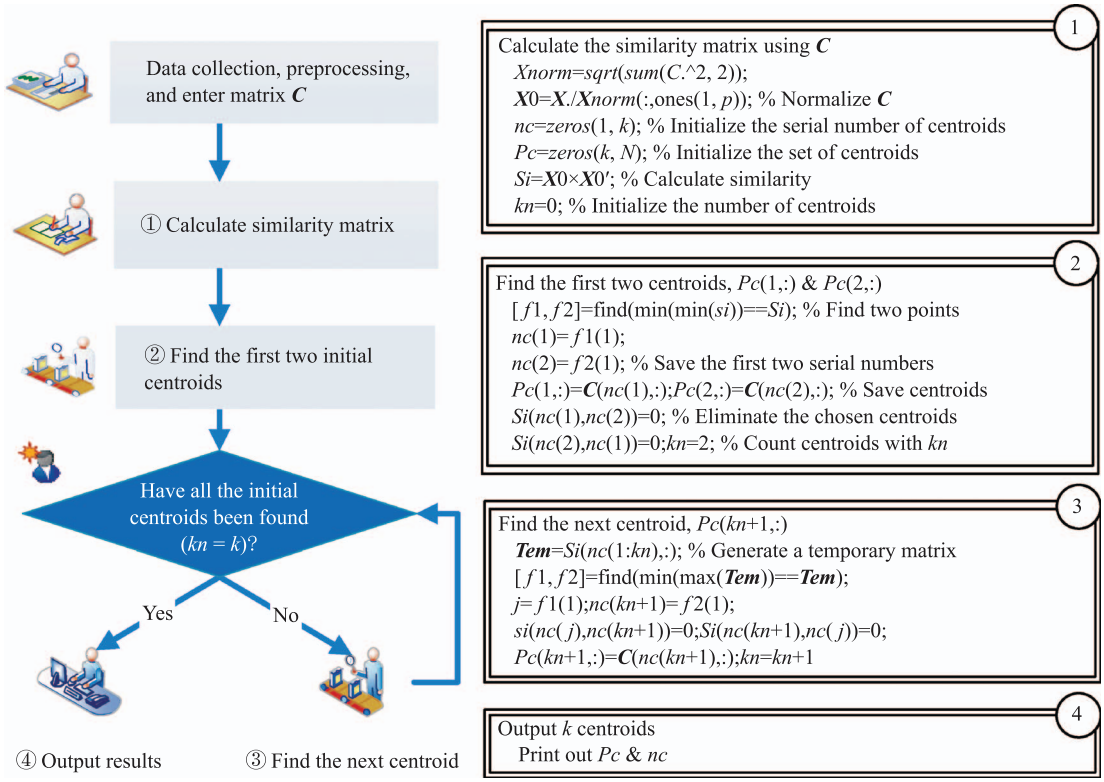


Fig. 1 Examining the k best initial centroids

For better understanding, here a brief example is given to show how to find the k best initial centroids.

There are 12 points in the plane right-angle coordinate system, and the matrix $\boldsymbol{X}$ gives the coordinate values of those points, which are shown in Fig. 2(a) and Fig. 2(b). Points are labeled according to their order in the matrix $\boldsymbol{X}$.

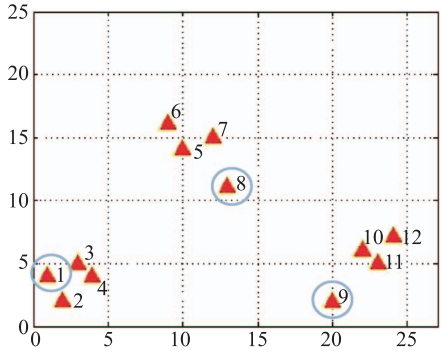
Steps followed are the detailed description of finding the k best initial centroids.

Step 1 For similarity comparison, normalize the coordinate values of points. Thus $\boldsymbol{X}0$ is obtained to calculate the similarity matrix, $\boldsymbol{PtP} = (\boldsymbol{X}0) * (\boldsymbol{X}0)'$, which are shown in Fig. 2(c).

Step 2 Find the first two initial centroids: find the minimum element in $\boldsymbol{PtP}$, and mark its row number as the first centroid and its column number as the second centroid. Shown as in Fig. 2, the 1st point and the 9th points are the first two initial centroids.

Step 3 Find the next initial centroid: from $\boldsymbol{PtP}$, extract rows corresponding to the existing centroids and change them into a matrix $\boldsymbol{Tem}$, and then find the minimum in $\max(\boldsymbol{Tem})$ and mark its column number as the next centroid, which is the 8th point here. At last, all the three centroids are found, shown in Fig. 2(a) and Fig. 2(d).

Note that in $\boldsymbol{Tem}$ the elements, relating to the existing centroids, change into 0, so the distractions are cleared in the step of finding the next centroid, shown in Fig. 2(d).



(a) Points shown in the coordinate system

$$\boldsymbol{X} = \begin{bmatrix} 1 & 4 \\ 2 & 2 \\ 3 & 5 \\ 4 & 4 \\ 10 & 14 \\ 9 & 16 \\ 12 & 15 \\ 13 & 11 \\ 20 & 2 \\ 22 & 6 \\ 23 & 5 \\ 24 & 7 \end{bmatrix}$$

(b) Points' coordinate values

$X0$=[0.242 5 0.970 1
0.707 1 0.707 1
0.514 5 0.857 5
0.707 1 0.707 1
0.581 2 0.813 7
0.490 3 0.871 6
0.624 7 0.780 9
0.763 4 0.645 9
0.995 0 0.099 5
0.964 8 0.263 1
0.977 2 0.212 4
0.960 0 0.280 0]

PtP=[1.000 0 0.857 5 0.956 7 0.857 5 0.930 4 0.964 5 0.909 1 0.811 8 0.337 9 0.489 3 0.443 1 0.504 5
0.857 5 1.000 0 0.970 1 1.000 0 0.986 4 0.963 0 0.993 9 0.996 5 0.774 0 0.868 2 0.841 2 0.876 8
0.956 7 0.970 1 1.000 0 0.970 1 0.996 8 0.999 6 0.991 0 0.946 5 0.597 3 0.722 0 0.684 9 0.734 0
0.857 5 1.000 0 0.970 1 1.000 0 0.986 4 0.963 0 0.993 9 0.996 5 0.774 0 0.868 2 0.841 2 0.876 8
0.930 4 0.986 4 0.996 8 0.986 4 1.000 0 0.994 2 0.998 5 0.969 3 0.659 3 0.774 9 0.740 8 0.785 8
0.964 5 0.963 0 0.999 6 0.963 0 0.994 2 1.000 0 0.986 8 0.937 2 0.574 6 0.702 3 0.664 2 0.714 7
0.909 1 0.993 9 0.991 0 0.993 9 0.998 5 0.986 8 1.000 0 0.981 3 0.699 3 0.808 1 0.776 3 0.818 4
0.811 8 0.996 5 0.946 6 0.996 5 0.969 3 0.937 2 0.981 3 1.000 0 0.823 9 0.906 4 0.883 2 0.913 7
0.337 9 0.774 0 0.597 3 0.774 0 0.659 3 0.574 6 0.699 3 0.823 9 1.000 0 0.986 2 0.993 5 0.983 1
0.489 3 0.868 2 0.722 0 0.868 2 0.774 9 0.702 3 0.808 1 0.906 4 0.986 2 1.000 0 0.998 6 0.999 8
0.443 1 0.841 2 0.684 9 0.841 2 0.740 8 0.664 2 0.776 3 0.883 2 0.993 5 0.998 6 1.000 0 0.997 6
0.504 5 0.876 8 0.734 0 0.876 8 0.785 8 0.714 7 0.818 4 0.913 7 0.983 1 0.999 8 0.997 6 1.000 0]

(c) Normalized X, $X0$ and similarity matrix PtP

Tem=[1.000 0 0.857 5 0.956 7 0.857 5 0.930 4 0.964 5 0.909 1 0.811 8 0.000 0 0.489 3 0.443 1 0.504 5
0.000 0 0.774 0 0.597 3 0.774 0 0.659 3 0.574 6 0.699 3 0.823 9 1.000 0 0.986 2 0.993 5 0.983 1]
max(Tem)=[1.000 0 0.857 5 0.956 7 0.857 5 0.930 4 0.964 5 0.909 1 0.823 9 1.000 0 0.986 2 0.993 5 0.983 1]

(d) Finding the 3rd centroid

Fig. 2 Example for finding the k best initial centroids

By choosing the initial centroids from a global optimization perspective, MMSK-means, on the one hand, avoids the local optimization trap, and on the other hand, reduces the iterations of the algorithm that are needed to make a global optimization choice. All these make the algorithm more stable.

4. Improved tag clustering based on LSA

4.1 Tag clustering based on LSA

LSA is a popular linear algebraic indexing method to produce low dimensional representations by word co-occurrence. It is an algebraic model of information retrieval, and a method for knowledge acquisition and knowledge display. As a theory of knowledge induction and representation, LSA is widely used in document retrieval research, by extracting and characterizing the words of context with the help of a large text corpus and statistical means. LSA is used to exterminate fuzzy problems in text data by determining the latent semantic structure between words after a statistical analysis of the abundant text, and then, LSA achieves its purpose of simplifying the text vectors by reducing the space dimensions. For example, LSA is applied to fully explore the semantic meaning of sparse tags in the study of less popular webpage classifications [43]. Additionally, in [44], LSA is used as the review and knowledge extraction methodology to research and systematically review the cross disciplinary literature.

LSA computes the orthogonal dimensions of a term-document matrix, and with SVD, which will be described later, LSA obtains a projection of the text in a lower dimensional subspace, based on which, the similarity of terms or documents can be calculated by counting the angles between space vectors after projection, and relation among terms and documents is then determined.

Any rectangular matrix, such as an $M * N$ matrix of terms and documents $\boldsymbol{C}$, whose rank should be r, can be decomposed by SVD as $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}$, where $\boldsymbol{S} = \mathrm{diag}(\lambda_1, \ldots, \lambda_r)$ and $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_r$ are singular values of $\boldsymbol{C}$. $\boldsymbol{U} = [u_1, \ldots, u_r]$ and u_i is called the left singular vector. $\boldsymbol{V} = [v_1, \ldots, v_r]$, where v_i is the right singular vector. Additionally, according to the low rank approximation, LSA uses the first k vectors in $\boldsymbol{U}$ as the transformation matrix to embed the original documents into a k-dimensional space, which is provided in Fig. 3.

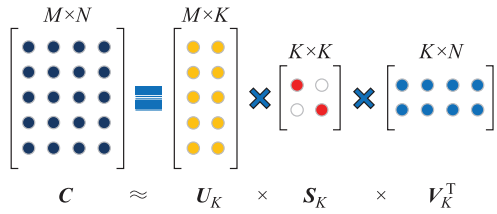


Fig. 3 SVD in LSA

In Fig. 3, both $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$ satisfy $\boldsymbol{U}_k^{\mathrm{T}}\boldsymbol{U}_k = \boldsymbol{I}$ and $\boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{V}_k = \boldsymbol{I}$. In the LSA research, the value of k is crucial. An ideal k meets the following: k is sufficiently large to make $\boldsymbol{U}_k * \boldsymbol{S}_k * \boldsymbol{V}_k^{\mathrm{T}}$ sufficiently similar with $\boldsymbol{C}$, and k is sufficiently small to ensure the eradication of redundant data and the reduction of dimensions.

For a thorough understanding of LSA, it is helpful to provide an explanation of practical significance for SVD. $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$, matrices of k-dimensional space, obtained by SVD from the original matrix of terms and documents $\boldsymbol{C}$, are used as the point coordinates of terms and documents, respectively. As k is the number of selected topics, each row of $\boldsymbol{U}_k$ is expressed as a term by the correlation degree

between the term and the topics, and the greater value of each dimension means that there is a stronger correlation between the term and the topic. Similarly, each row of $\boldsymbol{V}_k$ is expressed as a document by the correlation degree between the document and the topics, and the greater value of each dimension represents the greater importance of the document in that topic. In addition, $\boldsymbol{S}_k$ can be expressed as the correlation degree of terms and documents, whereas the larger the value is, the stronger the correlation of terms and documents is for that topic.

Thus, the method of clustering based on LSA is obtained. With $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$, to cluster terms or documents, for each row (a term or a document) researchers can select a topic in which the value in this dimension is the largest in the row. By this way, all of the terms or documents are classified into k clusters.

The tag clustering method based on LSA is provided as follows:

Step 1 Input the $M * N$ matrix of term-document $\boldsymbol{C}$.

Step 2 Do SVD of $\boldsymbol{C}$ using $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}$.

Step 3 Set the value of k: find the minimum that satisfies $\frac{\sum_{1}^{k} \lambda_i}{\sum_{1}^{rank(\boldsymbol{C})} \lambda_j} \geqslant \delta$.

Step 4 Do k-SVD of $\boldsymbol{C}$ and obtain k-dimensional coordinate of tags and resources using $\boldsymbol{C}_k = \boldsymbol{U}_k \boldsymbol{S}_k \boldsymbol{V}_k^{\mathrm{T}}$.

Step 5 Assign terms and documents to clusters using $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$ respectively.

Step 3, in which δ is set in advance, retains the similarity no less than δ and reduces the dimensions as much as possible. Additionally, the authors assign a term or a document to a cluster using the data obtained in Step 4. For example, if [0.1, 0.2, 0.9] is the second row of $\boldsymbol{U}_k$, which means k, the number of clusters is three, the correlation degree between the second term and the first cluster is 0.1, and the correlation degree between this term and the second cluster is 0.2, and the correlation degree between this term and the third cluster is 0.9. The second term is assigned to the third cluster, because their correlation degree is the biggest.

4.2 LMMSK: MMSK-means combined with LSA

It is noted above that the value of k should be set in advance before K-means. In this paper, k is set by $\frac{\sum_{1}^{k} \lambda_i}{\sum_{1}^{rank(\boldsymbol{C})} \lambda_j} \geqslant \delta$, called the cumulative importance inequality (CII) as introduced in the LSA-based algorithm (Step 3). Then, combining LSA with MMSK-means produces a new two-step social tag clustering method, LMMSK. Firstly, set the value of k by utilizing LSA. Secondly, input k and run MMSK-means, after which the k clusters are obtained. Elaborative descriptions are provided as follows:

Step 1 Input the $M * N$ matrix of tag-resource $\boldsymbol{C}$.

Step 2 Do SVD of $\boldsymbol{C}$ using $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}$.

Step 3 Set the value of k: find the minimum that satisfies $\frac{\sum_{1}^{k} \lambda_i}{\sum_{1}^{rank(\boldsymbol{C})} \lambda_j} \geqslant \delta$.

Step 4 Do k-SVD of $\boldsymbol{C}$ and obtain k-dimensional coordinate of tags and resources using $\boldsymbol{C}_k = \boldsymbol{U}_k \boldsymbol{S}_k \boldsymbol{V}_k^{\mathrm{T}}$.

Step 5 Classify social tags into k clusters utilizing $\boldsymbol{C}_k$ with MMSK-means.

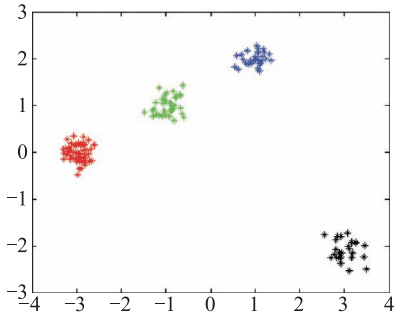
5. Experimental evaluation

Experiments are conducted in two stages, both of which are completed on Matlab: in the first stage, compare the clustering results of MMSK-means and the original K-means in Matlab, and MMSK-means becomes more stable and more accurate, and runs faster over fewer iterations; in the second stage, which is more crucial compared with the first stage, runs MMSK-means, the LSA-based algorithm and LMMSK with the same dataset, and compares the clustering results, determining that LMMSK performs the best.

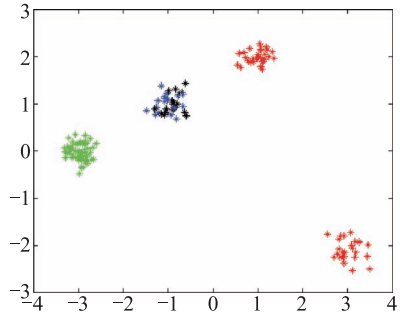
5.1 MMSK-means & K-means

With the same data, the authors run both MMSK-means and the original K-means in Matlab twice, and then compare the clustering results. There are six iterations the first time running K-means and five iterations the second time, whereas in both run of the MMSK-means, there are three iterations. This finding proves that MMSK-means reduces the time and space cost.

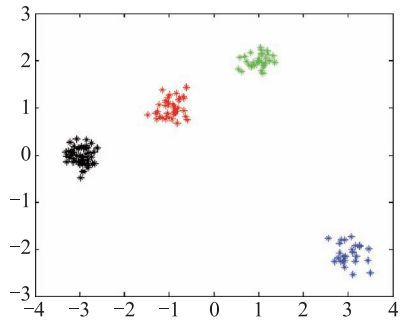
Fig. 4 provides the run results of both algorithms.



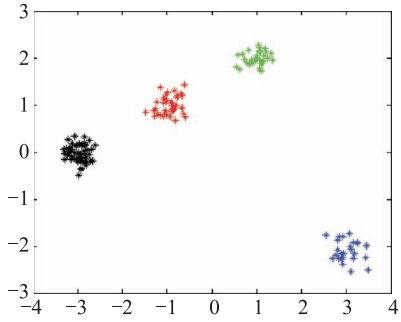
(a) First running result of K-means



(b) Second running result of K-means



(c) First running result of MMSK-means



(d) Second running result of MMSK-means

Fig. 4 Clustering results of K-means and MMSK-means

From Fig. 4, it is obvious that: (i) the running results of MMSK-means are consistent, although K-means runs unsteadily; (ii) the second running results of K-means are obviously incorrect: the first cluster and the fourth cluster from the top, which should be separated, are classified into the same cluster in red, and the second cluster from the top is divided into two clusters in black and blue, while MMSK-means runs well. Therefore, MMSK-means outperforms K-means.

5.2 LMMSK & MMSK-means & LSA

There are two tasks in the second stage: one compares LMMSK with MMSK-means, and the other compares LMMSK with LSA. The authors conduct experiments multiple times with the dataset, which is a real tag-resource dataset obtained from the Delicious Social Bookmarking System from 2004 to 2009. The original dataset, which was released in the framework of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011) at the 5th ACM Conference on Recommender Systems (RecSys 2011), contains social networking, bookmarking, and tagging information from a set of 2 000 users from the Delicious Social Bookmarking System (http://www.delicious.com). Its users are interconnected in a social network generated from Delicious "mutual fan" relations. Each user has bookmarks and tag assignments, i.e., tuples [user, tag, bookmark]. Each bookmark has a title and URL. Fig. 5 shows certain statistics regarding the dataset. The figure originates from the screenshot of a table from [45].

1 867 users
69 226 bookmarked URLs 38 581 bookmarked primcipal URLs (e.g.www.delicious.com for http://www.delicious.com; http://www.delicious.com/tag; http://www.delicious.com/help/api)
104 799 bookmarks (avg. 56.1 bookmarked URLs/user; avg. 1.5 users/bookmark)
53 388 tags 437 593 tag assignments (avg. 234.4 tas/user; avg. 6.3 tas/URL)
7 668 bi-directional user relations (avg. 8.2 relations/user)

Fig. 5 Data statistics of the hetrec-delicious-2k dataset

After the preliminary data processing, we obtain the tags and resource matrices from 2003 to 2009 as the objects to cluster. Although the data size from 2003, where there are 24 tags and 12 resources, is too small to cluster and regardless of whether δ is 0.2 or 0.25, the best number of clusters is 2. Therefore, we do not consider the data from 2003 when comparing and analyzing the experiments' results.

5.2.1 LMMSK & MMSK

The authors set δ to 0.25, while in MMSK, the number of clusters does not rely on δ. Thus in this comparison, the MMSK shares the same k with the LMMSK when their inputs are from the same year. A direct comparison is provided in Table 1.

The experimental results of running LMMSK and MMSK-means with tag-resource data from 2004 to 2009 are shown in Table 1. After standardization the total-cohesion and separateness are comparable here. The total-cohesion describes the summation of all within-cluster cohesions, and the larger the total-cohesion is, the better the clustering result will be. While the separateness describes the summation of distances between different centroids, and the bigger the separateness is, the better the clustering result will be. In addition, the two rows corresponding to the iterations are the number of iterations in running LMMSK and MMSK-means, and a clustering algorithm with less iterations saves time and space cost more.

Table 1 Total-cohesion & separateness & iteration of running LMMSK and MMSK-means

Results	Algorithm	2004	2005	2006	2007	2008	2009
Total-cohesion	MMSK	4.20	4.83	10.28	22.71	30.36	42.68
	LMMSK	55.60	113.45	1 021.42	2 735.26	6 567.08	12 049.26
Separateness	MMSK	2.69	2.71	2.57	2.51	2.58	2.65
	LMMSK	27.54	145.37	31 355.07	571 275	3 694 946	15 228 491
Iteration	MMSK	4	2	2	2	2	2
	LMMSK	2	2	2	2	2	2

In Table 1, the number of iterations in running LMMSK is slightly different with MMSK-means, and generally, the iterations in running LMMSK are no more than those in running MMSK-means. However, the total-cohesion and separateness are obviously different between the two algorithms. For example, in 2005, the total-cohesion of running MMSK-means is 4.828 427, whereas that of running LMMSK is 113.450 9, which means, in LMMSK, the cluster-inner points share greater cohesion than that in MMSK-means. And the result from every year shows that in LMMSK, the distinction between different clusters is more significant than that of MMSK-means. Thus, it is observed that LMMSK performs better.

5.2.2 LMMSK & the LSA-based algorithm

In the study of Zhang et al., experimental results have demonstrated that in text categorization, LSA has better performance than TF-IDF and multi-words [25]. As long as our comparison convinces that LMMSK performances better than LSA, LMMSK proves to be better than TF-IDF and multi-words.

In running LSA, there is no iteration. Thus in this section, the authors first compare the total-cohesion and the separateness between the results of LMMSK and LSA. The comparison is shown in Table 2.

Table 2 shows that the total-cohesion of LMMSK is greater than that of LSA, and simultaneously, the Separateness of LMMSK is far more than LSA. It seems that LMMSK outperforms LSA.

The truth is: comparing the experimental results of LMMSK and the LSA-based algorithm directly seems rude, because the two algorithms are totally different. While, in this section, the authors propose a CCR matrix, which is noted above as a clustering corresponding result matrix, to compare the experimental clustering results of LMMSK and the LSA-based algorithm, through the correspondence of every cluster of LMMSK with every cluster of the LSA-based algorithm.

Table 2 Total-cohesion & separateness of running LMMSK and LSA

Result	Algorithm	2004	2005	2006	2007	2008	2009
Total-cohesion	LSA	8.76	21.18	214.41	440.58	952.19	1 502.66
	LMMSK	55.60	113.45	1 021.42	2 735.26	6 567.08	12 049.26
Separateness	LSA	12.31	41.20	7230.00	71 294	348 747	1 109 107
	LMMSK	27.54	145.37	31 355.07	571 275	3 694 946	15 228 491

Table 3 shows the CCR matrix of LMMSK and the LSA-based algorithm with data from 2004, as an example to demonstrate how to compare the two algorithms. In Table 3, the Cluster-ID KC-series are the IDs of clusters in LMMSK, whereas the Cluster-ID LC-series are the IDs of clusters in the LSA-based algorithm, and the data in every cell correspond to the number of common points of the two clusters. For example, [2, 0, 4, 11], the first row, excluding the last number, is LC1, which is the first cluster of the LSA-based algorithm, whereas the first number "2" means that there are two points shared by LC1 and KC1, which is the first cluster of LMMSK, and the second number, "0", means that there is no point shared by LC1 and KC2, and so on. The last number, "17", means that the total number of points in LC1 is 17 (17=2+0+4+11).

Table 3 CCR matrix of LSA-LMMSK

Cluster-ID	KC1	KC2	KC3	KC4	Total
LC1	2	0	4	11	17
LC2	9	0	0	0	9
LC3	6	0	0	0	6
LC4	0	16	8	1	25
Total	17	16	12	12	57

Eliminating the last row and the last column, a cycle process will be used on the CCR matrix: find the largest data from the rows and the columns that are not eliminated, display it using bold and italics, and eliminate the row and the column where the data are located. This process continues until the largest number is 0. Thus, the greatest correspondence between the two algorithms is found, and by counting the ranges related to the correspondence, the authors

obtain the reciprocal accuracy (RA) of each algorithm. Below, an explanation is provided to obtain the reciprocal accuracy shown in Table 4.

Table 4 Obtaining the reciprocal accuracy of LSA & LMMSK

Cluster-ID	KC1	KC2	KC3	KC4	Total
LC1	2	0	4	⑪	17
LC2	⑨	0	0	0	9
LC3	6	0	0	0	-
LC4	0	⑯	8	1	25
Total	17	16	-	12	(45, 51)

For better explanation, the cycle process is concisely provided in Fig. 6.

Fig. 6 Brief demonstration of the method used to obtain the greatest correspondence

In Table 4, there are three numbers (16, 11, 9) in bold and italics, which means that they are selected by our cycle process. Counting the range in LMMSK, we find the related clusters, KC1 (17 points), KC2 (16 points), and KC4 (12 points), as well as the range (namely, the reciprocal accuracy, RA) is 45 (45=17+16+12). In the same way, it is easy to figure out that RA of the LSA-based algorithm is 51. It is observed that LMMSK has a better accuracy because the smaller the RA, the better the accuracy.

Above is an example of comparing LMMSK and the LSA-based algorithm only using data from 2004 and δ set as 0.20. Table 5 shows all of the RAs of LMMSK and the LSA-based algorithm after the cycle process of CCR matrices with data from 2004 to 2009 and δ set as 0.20 and 0.25.

Table 5 RAs of LMMSK and the LSA-based algorithm

δ	Algorithm	2004	2005	2006	2007	2008	2009
0.20	LSA	51	106	1 148	2 871	7 513	14 323
	LMMSK	45	96	886	2 531	5 995	11 717
0.25	LSA	54	112	1 143	2 914	7 414	14 321
	LMMSK	33	95	907	2 552	6 109	11 821

In Table 5, the figure in every cell represents the RA of the algorithm with the data from the corresponding year and δ. For example, in 2007, when δ is 0.20, the RA of the LSA-based algorithm is 2 871, whereas the RA of LMMSK is 2 531. From Table 5, it is obvious that no matter how much δ is or regardless of the year the data are from, the RA of the LSA-based algorithm is always larger than the RA of LMMSK, which means that LMMSK has a better accuracy compared with the LSA-based algorithm.

Finally, LMMSK turns out to be better than LSA. So, according to Zhang et al. [25], compared with TF-IDF and multi-words, LMMSK has better performance obviously.

5.3 Time consumption analysis

Apparently, there are two sequential steps in LMMSK, LSA and MMSK. When calculating the time consumption of LMMSK, T_{LMMSK}, the authors should calculate that of LSA and MMSK separately, and the bigger one is critical.

Considering an example: there are n points in p dimensions need to be clustered into k clusters. In traditional K-means, the time consumption is $O((n*p+k*p)*t)$, where t counts the iterations, $n*p$ indicates the time consumption of assigning points to the nearest centroids, and $k*p$ indicates the updating of centroids. The MMSK performs better, because the number of iterations is smaller. And after data processing, in LMMSK, the number of iterations even smaller than that in MMSK, so t changes to be a constant c. Thus in LMMSK, the time consumption of clustering is $T_c = O((n*p+k*p)*c)$.

However, the time consumption of LSA cannot be ignored, which is noted as T_{LSA} for convenience. When an $n*p$ matrix multiples a $p*p$ matrix, the time consumption is $O(n*p\wedge 2)$. In LSA, the main process is SVD, where an $n*p$ matrix is decomposed into the product of three matrices. Thus, there is no doubt that $T_{\mathrm{LSA}} > O(n*p\wedge 2)$. As we know, $T_{\mathrm{LMMSK}} = T_c + T_{\mathrm{LSA}}$, and because n is far more than k, it is obvious that $T_{\mathrm{LSA}} \gg T_c$. Therefore, $T_{\mathrm{LMMSK}} \approx T_{\mathrm{LSA}}$.

Although there is hardly any improvement in time consumption, at least LMMSK has better clustering results, compared with LSA. However, honestly, LMMSK gains better results compared with MMSK or K-means, at the cost time consumption.

6. Conclusions

This paper proposes MMSK-means first and then LMMSK with LSA. Experiments are conducted to confirm the improvement of MMSK-means compared with the original K-means, and describe the advantages of LMMSK over MMSK-means and the LSA-based algorithm on social tag clustering.

A few conclusions from this study are as follows:

(i) MMSK-means is capable of obtaining a global optimization and, in clustering, is stable compared with the original K-means. Moreover, MMSK-means reduces iterations and thus saves time and space cost.

(ii) LMMSK performs better than MMSK-means, likely due to the consideration of the semantic relation in tags, which makes clustering more directive and accurate. Fur-

thermore, the clustering results of LMMSK have a semantic association tendency and more practical significance due to the higher inner cohesion after LSA is introduced.

(iii) LMMSK has better accuracy than the LSA-based algorithm, likely because the former uses $\boldsymbol{C}_k$, which is the k-rank similar matrix of the original term-document matrix $\boldsymbol{C}$, as the clustering object, whereas the latter uses the tag coordinate set $\boldsymbol{U}_k$, which is obtained from $\boldsymbol{C} \approx \boldsymbol{U}_k \boldsymbol{S}_k \boldsymbol{V}_k^{\mathrm{T}}$ by SVD. The latter may ignore the influence of $\boldsymbol{V}_k$ (the relevance of resources and topics) and $\boldsymbol{S}_k$ (the importance of topics in tag-resource dataset).

The superiority of the improved method, LMMSK, has been testified by experiments. However, the semantic correspondence from the clusters to the topics cannot be captured directly, because a cluster is not assigned to a topic by the correlation degree, whereas it is in the LSA-based algorithm. In addition, using $\boldsymbol{C}_k$ as the input of MMSK-means in LMMSK helps to achieve better accuracy by counting the influence of $\boldsymbol{V}_k$ and $\boldsymbol{S}_k$, while it is difficult to estimate what and how much the influence is.

When comparing LMMSK and the LSA-based algorithm, the authors propose a new matrix, CCR, which is expected to be applied to capture the evolutions of the social tagging system. It is promising that the CCR can be an effective tool when researching knowledge diffusion or the evolution of a knowledge community. And by introducing LSA, LMMSK gains better clustering results compared with MMSK or K-means, at the time consumption. In the future, we will try a good way to reduce the time consumption when using SVD, like combing it with machine learning.

References

[1] S. Ghosh, A. Srivastava, N. Ganguly. Effects of a soft cut-off on node-degree in the Twitter social network. *Computer Communications*, 2012, 35(7): 784 – 795.

[2] K. Zolfaghar, A. Aghaie. Evolution of trust networks in social web applications using supervised learning. *Procedia Computer Science*, 2011, 3(1): 833 – 839.

[3] A. L. Traud, P. J. Mucha, M. A. Porter. Social structure of Facebook networks. *Physica A: Statistical Mechanics and its Applications*, 2012, 391(16): 4165 – 4180.

[4] J. Diederich, T. Iofciu. Finding communities of practice from user profiles based on folksonomies. *Proc. of the European Conference on Technology Enhanced Learning*, 2006: 288 – 297.

[5] C. M. A. Yeung, N. Gibbins, N. Shadbolt. A study of user profile generation from folksonomies. *World Wide Web Conference Series*, 2008, 356: 1 – 8.

[6] J. M. Chen, M. C. Chen, Y. S. Sun. A tag based learning approach to knowledge acquisition for constructing prior knowledge and enhancing student reading comprehension. *Computers & Education*, 2014, 70(1): 256 – 268.

[7] M. Wang, B. Ni, X. S. Hua, et al. Assistive tagging: a survey of multimedia tagging with human-computer joint exploration. *ACM Computing Surveys*, 2012, 44(4): 1173 – 1184.

[8] M. Guy, E. Tonkin. Folksonomies: tidying up tags. *D-lib Magazine*, 2006, 12(1): 7 – 20.

[9] C. Held, J. Kimmerle, U. Cress. Learning by foraging: the impact of individual knowledge and social tags on web navigation processes. *Computers in Human Behavior*, 2012, 28(1): 34 – 40.

[10] K. Sun, X. Wang, C. Sun, et al. A language model approach for tag recommendation. *Expert Systems with Applications*, 2011, 38(3): 1575 – 1582.

[11] S. Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 2010, 12(10): 2011 – 2024.

[12] M. J. Barber, J. W. Clark. Detecting network communities by propagating labels under constraints. *Physical Review E*, 2009, 80(2): 283 – 289.

[13] G. Cordasco, L. Gargano. Community detection via semi-synchronous label propagation algorithms. *Proc. of the IEEE International Workshop on Business Applications of Social Network Analysis*, 2010: 1 – 8.

[14] R. Krestel, P. Fankhauser, W. Nejdl. Latent dirichlet allocation for tag recommendation. *Proc. of the Third ACM Conference on Recommender Systems*, 2009: 61 – 68.

[15] B. Sigurbjörnsson, R. Van Zwol. Flickr tag recommendation based on collective knowledge. *Proc. of the 17th International Conference on World Wide Web*, 2008: 327 – 336.

[16] J. Chen, S. Feng, J. Liu. Topic sense induction from social tags based on non-negative matrix factorization. *Information Sciences*, 2014, 280(1): 16 – 25.

[17] M. Mahajan, P. Nimbhorkar, K. Varadarajan. The planar K-means problem is NP-hard. *Proceedings of WALCOM: Algorithms and Computation*, 2009, 442(8): 274 – 285.

[18] M. J. Li, M. K. Ng, Y. M. Cheung, et al. Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters. *IEEE Trans. on Knowledge and Data Engineering*, 2008, 20(11): 1519 – 1534.

[19] J. Magidson, J. Vermunt. Latent class models for clustering: a comparison with K-means. *Canadian Journal of Marketing Research*, 2002, 20(1): 36 – 43.

[20] R. J. Kuo, S. Y. Lin, C. W. Shih. Mining association rules through integration of clustering analysis and ant colony system for health insurance database in Taiwan. *Expert Systems with Applications*, 2007, 33(3): 794 – 808.

[21] R. J. Kuo, H. S. Wang, T. L. Hu, et al. Application of ant K-means on clustering analysis. *Computers & Mathematics with Applications*, 2005, 50(1012): 1709 – 1724.

[22] Y. Zhao, G. Karypis, U. Fayyad. Evaluation of hierarchical clustering algorithms for document datasets. *Proc. of the 11th International Conference on Information and Knowledge Management*, 2002: 515 – 524.

[23] K. J. Han, S. Kim, S. S. Narayanan. Strategies to improve the robustness of agglomerative hierarchical clustering under data source variation for speaker diarization. *IEEE Trans. on Audio, Speech, and Language Processing*, 2008, 16(8): 1590 – 1601.

[24] C. Ambroise, G. Sèze, F. Badran, et al. Hierarchical clustering of self-organizing maps for cloud classification. *Neurocomputing*, 2000, 30(1/4): 47 – 52.

[25] W. Zhang, T. Yoshida, X. Tang. A comparative study of TF-IDF, LSI and multi-words for text classification. *Expert Systems with Applications*, 2011, 38(3): 2758 – 2765.

[26] S. C. Deerwester, S. T. Dumais, T. K. Landauer, et al. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 1990, 41(6): 391 – 407.

[27] T. Hofmann. Probabilistic latent semantic indexing. *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999: 50 – 57.

[28] A. Popescul, D. M. Pennock, S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. *Proc. of the 17th Conference on Uncertainty in Artificial Intelligence*, 2001: 437 – 444.
[29] Y. T. Zhang, L. Gong, Y. C. Wang. An improved TF-IDF approach for text classification. *Journal of Zhejiang University Science A*, 2005, 6(1): 49 – 55.
[30] J. Gemmell, A. Shepitsen, B. Mobasher, et al. Personalizing navigation in folksonomies using hierarchical tag clustering. *Proc. of International Conference on Data Warehousing and Knowledge Discovery*, 2008: 196 – 205.
[31] C. Hayes, P. Avesani. Using tags and clustering to identify topic-relevant blogs. *Proc. of the 1st International Conference on Weblogs and Social Media*, 2007: 1 – 8.
[32] M. Girvan, M. E. Newman. Community structure in social and biological networks. *Proc. of the National Academy of Sciences*, 2002, 99(12): 7821 – 7826.
[33] M. E. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 2006, 103(23): 8577 – 8582.
[34] A. L. Barabási. Scale-free networks: a decade and beyond. *Science*, 2009, 325(5939): 412 – 413.
[35] A. L. Barabási, H. Jeong, R. Ravasz, et al. On the topology of the scientific collaboration networks. *Physica A*, 2002, 311: 590 – 614.
[36] J. Leskovec, J. Kleinberg, C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. *Proc. of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005: 177 – 187.
[37] L. Backstrom, D. Huttenlocher, J. Kleinberg, et al. Group formation in large social networks: membership, growth, and evolution. *Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006: 44 – 54.
[38] T. Falkowski, J. Bartelheimer, M. Spiliopoulou. Mining and visualizing the evolution of subgroups in social networks. *Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence*, 2006: 52 – 58.
[39] G. Begelman, P. Keller, F. Smadja. Automated tag clustering: improving search and exploration in the tag space. *Proc. of Collaborative Web Tagging Workshop*, 2006: 15 – 33.
[40] E. Giannakidou, V. Koutsonikola, A. Vakali. Co-clustering tags and social data sources. *Proc. of the 9th International Conference on Web-Age Information Management*, 2008: 317 – 324.
[41] R. Forsati, M. Mahdavi, M. Shamsfard, et al. Efficient stochastic algorithms for document clustering. *Information Sciences*, 2013, 220(1): 269 – 291.
[42] C. D. Nguyen, K. J. Cios. GAKREM: a novel hybrid clustering algorithm. *Information Sciences*, 2008, 178(22): 4205 – 4227.
[43] J. Wang, J. Peng, O. Liu. A classification approach for less popular webpages based on latent semantic analysis and rough set model. *Expert Systems with Applications*, 2015, 42(1): 642 – 648.
[44] A. Kundu, V. Jain, S. Kumar, et al. A journey from normative to behavioral operations in supply chain management: a review using latent semantic analysis. *Expert Systems with Applications*, 2015, 42(2): 796 – 809.
[45] I. Cantador, P. Brusilovsky, T. Kuflik. Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011). *Proc. of the 5th ACM Conference on Recommender Systems*, 2011: 387 – 388.

Biographies

Jing Yang was born in 1992. She received her B.S. degree from Department of Information Systems in School of Economics and Management, Beihang University. She is a graduate student of Beihang University. Her research interests include knowledge management, data mining and complex networks.
E-mail: Jing_Y1992@hotmail.com

Jun Wang was born in 1969. He received his Ph.D. degree in management sciences from Northeastern University, Shenyang, China, in 2003. He is currently a professor in Department of Information Systems, Beihang University, Beijing, China. He is the author or coauthor of more than 20 papers published in international journals. His current research interests include knowledge management, knowledge systems engineering, business intelligence and decision analysis.
E-mail: king.wang@buaa.edu.cn