

ALMA MATER STUDIORUM · UNIVERSITÀ DI
BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Magistrale in Scienze di Internet

**AUTOMATED CLUSTERING IN
COLLABORATIVE TAGGING
SYSTEMS: SCOPES AND
METHODS**

Tesi di Laurea in Microeconomia e Teoria dei Giochi

Relatore:
Chiar.mo Prof.
GIOVANNI ROSSI

Presentata da:
YANLONG ZHANG

I Sessione
Anno Accademico 2010-2011

For my husband...

Indice

Introduzione	1
1 Collaborative Tagging System	4
1.1 Definition	4
1.2 Background	5
1.3 Folksonomy And Taxonomy	6
1.3.1 Metadata	6
1.3.2 Taxonomy	7
1.3.3 Folksonomy	9
1.4 Delicious	10
2 Why Social Tagging?	14
2.1 Motivation	14
2.2 Advantage	14
2.3 Social Intelligence	16
2.4 Limitation	19
2.4.1 Semantic Ambiguity	19
2.4.2 Formative Drawbacks	21
2.4.3 Limited Search	22
2.4.4 Solution	22

3	Preliminary of Clustering	24
3.1	Scopes of Clustering	24
3.1.1	Query Handling	24
3.1.2	Organization of Resource Collection	25
3.1.3	Overcoming Ambiguities	25
3.1.4	Recommendation	26
3.2	Clustering	26
3.3	Similarity	28
3.3.1	Distance	28
3.3.2	Similarity	29
3.4	Types of Clustering	30
3.4.1	Hierarchical Clustering	30
3.4.2	Partitional Clustering	31
3.4.3	Spectral Clustering	31
3.4.4	Hard Clustering and Fuzzy Clustering	31
4	Clustering Methods In Collaborative Tagging Systems	33
4.1	Structure of Tagging Systems	33
4.2	Hierarchical Clustering	34
4.3	K-means Clustering	39
4.4	Relational structure and graphs	41
4.4.1	Preparing Dataset	42
4.4.2	Strongly Related Tags	42
4.4.3	Similarity	43
4.4.4	Weighted Graph Creation	44

4.4.5	A Clustering Algorithm	46
4.5	Fuzzy Clustering	55
4.6	Time sensitiveness	56
5	Recommendation In Collaborative Tagging Systems	58
5.1	Personalized Recommendation	59
5.2	Tag Recommendation	62
	Conclusion	63
	Bibliography	66

Introduzione

Questa tesi si occupa di clustering nei collaborative tagging systems, con particolare attenzione per gli algoritmi presenti in letteratura.

Visto lo sviluppo di internet di questi ultimi anni, Web 2.0 è un importante strumento per individuare quelle applicazioni online che permettono l'interazione fra sito e utente. Il tagging è uno dei servizi caratteristici di Web 2.0; esso permette agli utenti di classificare collaborativamente risorse e/o di trovare informazioni sulla base della classificazione stessa. Nel complesso, un collaborative tagging system comprende il processo attraverso il quale gli utenti associano tags, in forma di metadato, ai contenuti web, così da condividere questi ultimi. Essenzialmente, si tratta di un metodo collaborativo per creare e gestire i tags in termini dei quali vengono automaticamente catalogate le risorse.

Una caratteristica importante dei collaborative tagging systems è l'approccio sociale su cui si basano: sono solitamente creati da un insieme di individui o utenti (interessati alle risorse che vengono catalogate), che associano tags ai contenuti (ad esempio immagini, video e testi) adottando termini del linguaggio comune. I tags sono quindi creati dagli stessi utilizzatori nel momento in cui questi ultimi decidono quale parola (intesa in senso lato, cioè come sequenza di caratteri) associare al generico contenuto che stanno consultando. Delicious costituisce probabilmente l'esempio fondamentale di collaborative tagging system, utilizzato da un'ampia popolazione di utenti per l'archiviazione, la ricerca e la condivisione di contenuti web. Essendo intrinsecamente determinato dal tagging dagli utenti, il sistema riflette quindi in maniera

diretta quali parole sono più popolari, la diversità nei modi di pensare degli utenti, così come anche gli interessi di questi ultimi per le risorse.

Una sorta di intelligenza sociale (social intelligence in letteratura) viene raggiunta quando la popolazione di utenti è ragionevolmente grande e qualificata. In questo caso il collaborative tagging system fornisce una piattaforma per la condivisione della conoscenza professionale ed il miglioramento dell'organizzazione delle risorse. In generale, questi sistemi collaborativi sono in grado di adattarsi rapidamente al cambiamento di vocabolario, opinioni condivise e risultati disponibili (si pensi all'ambito medico, per esempio). Gli utenti possono condividere o scoprire le risorse attraverso la rete collaborativa e per questa via anche connettersi ad altri utenti con interessi simili ai loro.

In questi sistemi collaborativi, una funzione principale è il recupero delle informazioni (information retrieval). Selezionando un tag precedentemente utilizzato, è facile recuperare le risorse associate a quel tag. In ogni caso il tagging emerge da un comportamento spontaneo e libero che risulta in una grande varietà di tag utilizzati. In quest'ottica, l'ambiguità del generico tag (che si ha quando diversi tags hanno lo stesso significato oppure un singolo tag ha molti significati diversi) può falsamente dare l'impressione che alcune risorse siano simili fra loro anche quando esse sono in realtà molto diverse. Questo problema può generare difficoltà nel misurare la similarità tra le risorse per il recupero di informazioni, così come può anche generare ridondanza di tags.

In questo lavoro ci si occupa con particolare attenzione delle tecniche di data mining, e in particolare di clustering, automatico su grandi quantità di dati, con il fine di investigare se un uso opportuno di queste tecniche può consentire i suddetti problemi di ridondanza e ambiguità dei tags nei collaborative systems. Un algoritmo di clustering consente di dividere i tags in gruppi o sottoinsiemi basandosi su una misura di distanza oppure di similarità tra due tags. Il risultato, che è tecnicamente una partizione dei tags (presenti al generico istante) in blocchi o clusters, consente di gestire efficientemente le

query, riportando agli utenti le risorse più interessanti per ogni ragionevole insieme di tag inseriti per la ricerca.

Nel capitolo 1 si introducono i collaborative tagging systems in termini generali, confrontandoli con la tassonomia tradizionale. Si descrive poi più in dettaglio il sistema Delicious e alcuni suoi utilizzi.

Nel capitolo 2 si analizzano i vantaggi dei collaborative tagging systems, e si presentano poi le limitazioni nella navigazione del sistema e nel recupero d'informazione, dovute ad esempio alle ambiguità semantiche dei termini e alla limitazione della ricerca.

Nel capitolo 3 si considerano le motivazioni per l'utilizzo di algoritmi di clustering, introducendo brevemente il processo di clustering in generale, la misura di similarità ed il funzionamento dei diversi tipi di algoritmi.

Nel capitolo 4 si discutono in particolare tre algoritmi di clustering utilizzabili nei collaborative tagging systems, ovvero il clustering gerarchico (hierarchical clustering), il K-means clustering, e il clustering spettrale (spectral clustering).

Nel capitolo 5 si illustrano le ragioni per cui è necessario eseguire il tag clustering nei sistemi collaborativi: utilizzando insiemi di tags come termini di ricerca, si possono raccogliere informazioni sugli interessi degli utenti, eseguire efficacemente il recupero delle informazioni (information retrieval) e gestire le query fornendo in risposta le risorse disponibili più adeguate.

Chapter 1

Collaborative Tagging System

1.1 Definition

Tag

A tag is a user-contributed metadata, providing a mean of information or content item, created freely by users with personally salient keywords or labels, known as tags. The process of labeling is called tagging. Users can re-find the information later by means of those tags that they have created[4]. Also,by tagging users can store resources for their future retrieval.

Collaborative tagging system

A collaborative tagging system is a classification procedure and a collaborative method to create and manage tags and categorize resources. It describes a process by which many users add tags to share web resources[11], and is also known as «social classification», «social tagging» and «folksonomy».

Folksonomy is a popular concept describing collaborative tagging as a creative process coined by Thomas Vander Wal. It is a combination of folk and taxonomy, like "a people's taxonomy".

Collaborative tagging is a classification by the users and for the users. It is a social, decentralized and complex network where many annotations, generally provided by interrelated groups of individuals, are organized so to link resources and tags. Each resource item can be associated with many different tags, rather than with a single branch of a hierarchy. With tags chosen freely from common language and associated with web resources that are interesting for users (such as photographs, videos, web links and documents), collaborative tagging offers a sense of community in managing resources and results in a process of knowledge construction. Users can share their resources with others, discover resources through the collaborative network, and contact people with similar interests. The benefit of collaborative tagging systems comes from the many views of the mass, rather than from a dominant opinion supplied by a few.

1.2 Background

With the advent of the Internet, it becomes constantly easier to use digital networks for working informally as part of a community.

In 1990, people started to add keywords to documents and articles, which were text submitted to digital libraries. This allows people to organize documents in their collections by the keywords assigned. With this method, the indexing or classification is made by an authority, like a librarian, or results from the material supplied by the authors of the documents.

In the late 1990s, the term «blog» has been introduced by Peter Merholz. It was initially thought as a short form for weblog. It is an application of online diary consisting of a title, a time of publication, a body of article, and usually assigned to one or more categories or tags. The metadata created and entered by users became popular from then on.

In 2003, Delicious, an online bookmark manager, was founded by Joshua Schacter. It allows a user to store and share bookmarks, and add tags using

a non-hierarchical keyword classification. It also allows a user to see other users' tags thereby frequently finding different objects tagged by others.

After Delicious, collaborative tagging systems have been quickly replicated by other social applications, such as Flickr (another popular tagging system allowing users to upload, share and annotate images), YouTube (allowing users to tag their video collection, and uses the wisdom of crowds to generate recommendations).

From then on, the collaborative creation of tags by individual users, in a free form, is known as "folksonomy", and in 2004 it became the main characteristic of Web 2.0.

1.3 Folksonomy And Taxonomy

1.3.1 Metadata

Marking a content item with keywords or tags in the form of metadata is a common way to organize content for navigation, filtering or searching in the future[1]. Metadata is often referred to as "data about data", providing information about the data[11]. These information, often with high structure (related to documents, books, articles, photographs or other resources), were designed to support specific functions used to help in data organization and access.

There are many specify types of metadata[11] :

- Structural metadata: used to describe tables, indexes and other structure items of computer systems;
- Guide metadata: is usually expressed as a set of keywords to help human when searching for specific contents;

- Descriptive metadata: is used to search objects, such as titles, authors, subjects, keywords; it is used for organizing information according to its intellectual content.

In order to understand the creation of metadata, we need to know who produces these labels or tags for exploring resources available on the Internet. When introducing the notions of tagging and folksonomy, Mathes[2] summarizes three different ways of creating metadata according to its generator: professional, author or user. Wolfgang[9] also considers that interpreters (or information experts) label documents with the aid of ontologies, thesauri or classification systems to highlight the relation between concepts; on the other hand, authors also add remarks to their articles; finally, users interpret the document's content and attach tags taken from their occupational area

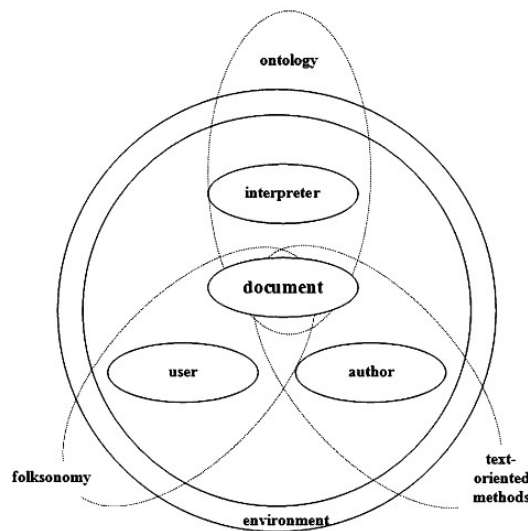


Figure 1.1: the creation of metadata by professionals, authors or users

1.3.2 Taxonomy

Professionals' metadata

The method for creating metadata is conceived by dedicated professionals, with suitable education and training. Typically, catalogers create metadata for digital libraries or other institutions online, where these latter developed sophisticated rules and schemes for cataloging[2].

For example, in the digital library all items (including books and magazines) are stored in a catalog. Each resource is usually stored in a precise catalog with some metadata about the author, a primary category, a secondary category and some related keywords. The categorization in catalogs is based on hierarchical and rigorous classification systems, with the keywords controlled by thesaurus. In general, metadata created by professionals are considered of high quality, but need costly time and effort to be produced, especially when a large amount of new content need to be created or cataloged.

Authors' metadata

Another approach for generating metadata is provided by authors, who are the original creators of the material that require metadata along with their works[2].

Both these two methods are used by the taxonomy system, which is classified and organized in a hierarchical, highly rule-oriented, and exclusive structure. This results in the following problem: there are users excluded from the metadata generating process who are still intended to be capable of providing valuable information. Typically, this hierarchical structure is organized by generalization-specialization relationships, also called informally «parent-child» relationships.

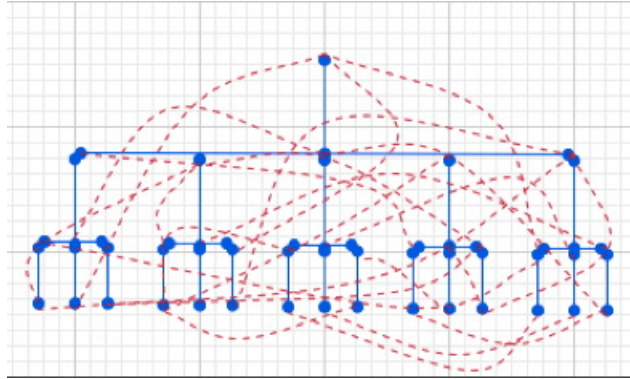


Figure 1.2: hierarchical structure

As shown in figure 1.2, there is a top level and several lower subdirectories, containing in turn further subdirectories all the way down. In this system, each item is assigned to only one specific category. Although there are too many folders in a hierarchy, (especially some of them are created by chance), a hierarchical structure can organize files accurately and unambiguously, while also bounding the contents of a folder. Unlike keyword- or tag-based search, through which seekers cannot be sure that a query has returned all the relevant informations, a hierarcal folder assures that all files contained in it are in a stable position.

1.3.3 Folksonomy

User-created metadata, known as tagging, this is another approach which is becoming a most popular method to organize and search content. It appears to be very competitive with respect to conventional classification and original metadata. In contrast with the taxonomy moder described above, in folksonomy there is no hierarchy or relationship specified such as a parent-child relationships between tags. It is a flat structure or unstructured system of metadata, where all tags are equally important[2].

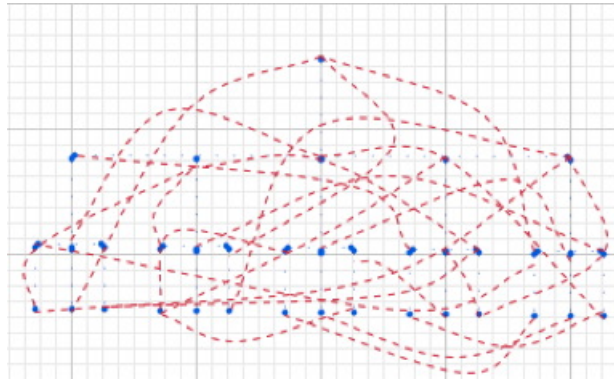


Figure 1.3: folksonomy

In this system, several users may simultaneously attach distributed, unstructured metadata to a document or other resources online. They can use a wide variety of tags, both general and specific types, to express their interest in resources. These types of tags, serve both as keywords and “classes”.

Thanks to the flat structure, it is possible to find related tags which may have been separated by long-distances in a taxonomy.

1.4 Delicious

Delicious is one of the most popular collaborative tagging system for web bookmarking. Its founder Joshua Schachter calls it "a social bookmarks manager"[3]. The organization of bookmarks is based on folksonomies, where users attach distributed tags for their favorite websites. In fact, Delicious is a service platform, operates as an architecture of participation, in which users add value to the application as a result of usage[13].

Bookmarks and tags

For personal use, users bookmark or tag (interpreted as a verb) because they want to keep in touch with interesting web pages: they can easily save a page with a bookmark on the remote web server of Delicious, in the same

way as they store bookmarks and favourites in their browser on computers. This application has a main practical benefit: once bookmarks are saved, they are accessible from anywhere and from any computer, not just from only one specific browser. This is useful when a person needs to use different computers, at home, at school or at work, which is a key feature of Delicious. Once users have created an account, they can start bookmarking websites with tags. Bookmarks and tags can be saved as 'private', namely accessible only by its tagger, or else as 'public', namely visibly to all users. This means that tags can be shared with others. As shown in Figure 1.4, it is a personal organization of one's own data.

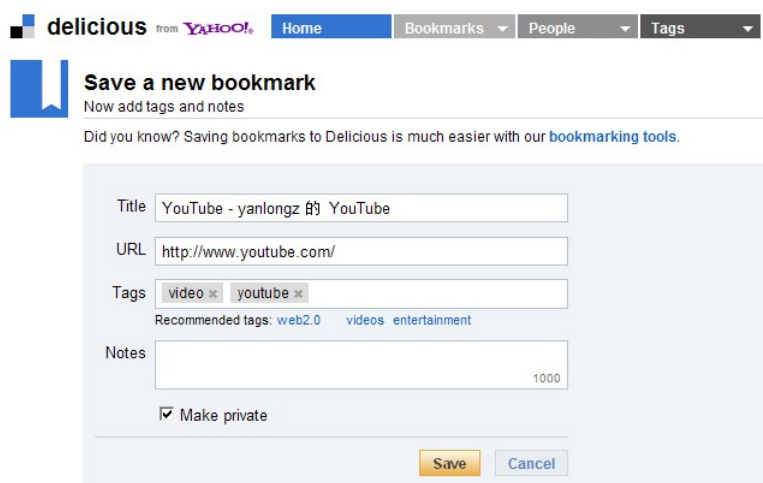
The image shows a screenshot of the Delicious website's 'Save a new bookmark' form. At the top, there is a navigation bar with the Delicious logo (a blue square with a white bookmark icon) and the text 'delicious from YAHOO!'. To the right of the logo are four menu items: 'Home', 'Bookmarks', 'People', and 'Tags', each with a small downward arrow. Below the navigation bar, the main heading is 'Save a new bookmark' in bold, followed by the sub-heading 'Now add tags and notes'. A small blue bookmark icon is to the left of the heading. Below the heading, there is a line of text: 'Did you know? Saving bookmarks to Delicious is much easier with our [bookmarking tools](#).' The form itself is a light blue box containing several input fields: 'Title' with the text 'YouTube - yanlongz 的 YouTube', 'URL' with 'http://www.youtube.com/', 'Tags' with 'video' and 'youtube' (each followed by a small 'x' icon), and 'Notes' with a large empty text area and a '1000' character limit indicator. Below the 'Notes' field is a checkbox labeled 'Make private' which is checked. At the bottom right of the form are two buttons: 'Save' (yellow) and 'Cancel' (grey).

Figure 1.4: bookmark creation

Each bookmark is based on an URL, which is a link to a web page associated with the following metadata: the title indentifying a web page, and the tags representing personal opinions. It allows users to choose freely any vocabulary to tag, as well as to jam more than one word together for describing page contents.

Every user has a personal page <http://del.icio.us/username> displaying one's own bookmarks. All bookmarks are automatically displayed in reverse chronological order, together with the list of all tags created by the user. By select-

ing a tag, one can filter his own bookmark space to view only those resources with that chosen tag.

Tag view

Delicious allows multiple users to save the same webpage, and lets them label it with different tags. Hence, looking at others' tags and corresponding resources is another feature of this application. There are two main ways to view social tagging. One is browsing tags from the tag page, where this latter displays recently added tags, together with bookmarks and a popular tag list such as that on the right side of Figure 1.5.

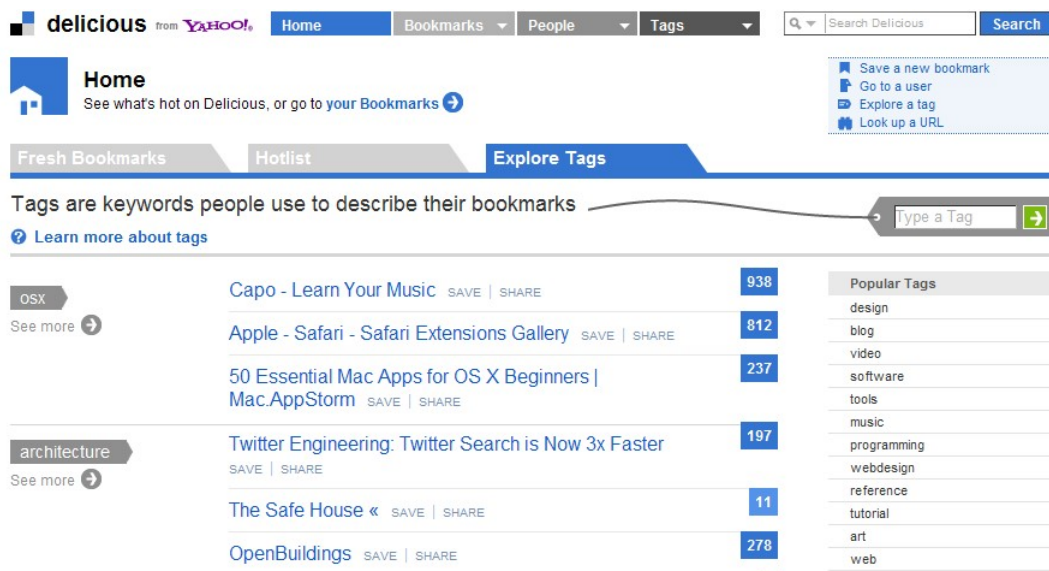


Figure 1.5: tag page

Another way is viewing tags as a tag-cloud, that is, a set of tags where their size reflects frequency and popularity. More precisely, most used tags are shown in biggest font and less used ones are shown in a smaller one.

In fact, there are two types of users in collaborative tagging systems: taggers and seekers.

- The former organize bookmarks and make them easier to re-find in the future.
- The later navigate seeking information; through tags they can find useful resources and people with common interests.

From the users' viewpoint, navigation through a collaborative tagging system is similar to implementing a keyword-based search. Clicking a tag from a tag list or tag-cloud works like entering a filtering system: out of the documents previously tagged, the system returns only those items annotated with that specific tag. Depending on the query, a collaborative tagging offers a method that assembles the union of tags, instead of providing their intersection, so it returns all the elements of any chosen tags[3].

Users who find interesting resource through queries are also likely to find other users with common interests. In fact, Delicious provides a web feed for users' lists of resources, organized by tags. This allows subscribers to be aware of new informations that are saved, shared and tagged by other users[11].

Although Delicious is neither unique nor pioneering in the field of bookmark management, its novelty and diversity (allowing users to add tags as a organizational construct) made it mostly popular.

Chapter 2

Why Social Tagging?

2.1 Motivation

Ames and Naaman have studied the reasons for tagging and found that tags were used for both organization and to communication, and were used both for selfish and social purposes[39].

- For selfish or personal purposes, the tagging system allows users to develop a personal digital filing system that applies keywords and easily retrieves the annotated resources.
- For social communication, it allows users to express themselves and share resources with other system users. It reflects a common interest and provides a tool for searching informations collected by multiple users.

2.2 Advantage

Taking Delicious as a specific example of collaborative tagging system, now we can see clearly the advantages of this folksonomy-based method over traditional classification.

	taxonomy	folksonomy
Producer	Professional	Users of internet
Maintenance cost	High	Low
Update cycle	Long cycle	Update immediately
Normativity	Normative and rigorous	Free tagging
Convenience to change	Complex	Simple and convenient
Descriptive ability	Unilateral and limited description	wisdom of crowds for tagging
Communication	Little	social communication

Table 2.1: folksonomy vs. taxonomy

As already mentioned, comparison table 2.1 above outlines how the taxonomy approach is hierarchical and rigorously rule-oriented. It requires to set up a costly separate department of experts dedicated to developing and maintaining the classification system. In contrast, the folksonomy approach is much simpler and cheaper, and displays further advantages listed hereafter.

- Firstly, the folksonomy approach is rapidly adaptable to changing vocabularies. Tagging is a simple way to manage informations. One can add, change or remove a tag when its meaning varies over time. The system updates and reflects immediately the results of such changes without any need to wait for a long as required, instead by taxonomy maintenance.
- Secondly, as a collaborative system allows users to freely add tags and participate in the process of content classification (without the effort involved by adding terms to a controlled vocabulary[2], which would constrain the action of individuals), users can use a wide variety of vocabularies, more diversified than those of classical taxonomies. The study of tagging usage in delicious shows that several tags perform different functions in addition to bookmarking. For example, the identify what is the bookmark and what it is about, who owns it, and even

the category and characteristics of the content. Some of these tags are useful only for their creator, as "unread" or "favorite", while others are useful for the public.

- Lastly, the flat and distributed structure of collaborative tagging systems appears to be a main advantage. In fact, a limitation of the taxonomy approach is that each element must be assigned to a single primary category, although they can be assigned to one or more secondary categories. Golder and Huberman[3] provide di example of an article about cats in Africa: when using a hierarchical system, the article is assigned to the sub-category "cats" in the category "Africa" (or the converse), but they cannot be assigned to one category at the same time. In collaborative tagging systems, however, this is possible because the tag space is flat and all tags are equally important.

2.3 Social Intelligence

A feature of any collaborative system is its inclusiveness, which yields rich user profiles. Like users' tagging, the system also directly reflects popular words, diversity in ways of thinking as well as in interests about resources, regardless of any common viewpoint, background, and prejudice. It can therefore be perceived as a democratic system, where everyone has the chance to contribute and share keywords as well as opinions. "The value in this external tagging is derived from people using vocabulary and adding explicitly their own meaning, which may come from inferred understanding of the information / object." Vander Wal (2005).

Collaborative tagging offers the possibility of organizing information by means of collective knowledge, provided by a community of users. A social intelligence is achieved when a mass of people participate within the system. Because of the great homogeneity in users' interest and knowledge, collaborative systems provide a platform for sharing valuable information as well as for improving

efficiently the organization of web resources. They are also becoming an important tool for information retrieval. In particular, Delicious has the declared intent to aid retrieval by finding common tags automatically. When a user finds his interested resources, he may save and tag the resources for his personal collection, which provides a feedback cycle[19]. This is an important characteristic for tagging system.

Common tags

Although tags describe different resources and are provided by different users, they have been shown to converge over time to a stable power law distribution, with clear correlations between tags. This enables to visualize collaborative tagging as follows.

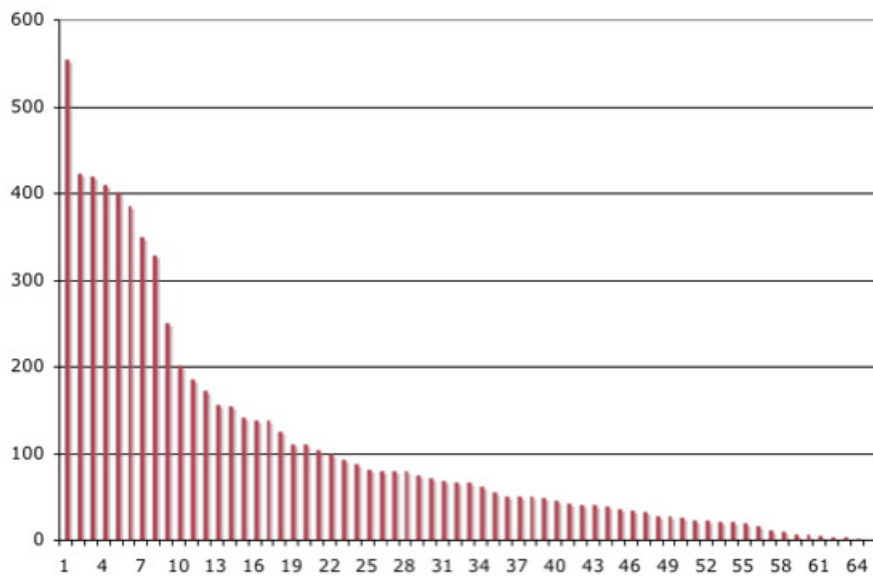


Figure 2.1: power law of tagging

The power law distribution of tagging in Delicious displays a regular pattern, with tagging behaviors appearingly stable and converging. As the system allows for many users to store the same page and potentially associate with

it different tags[14], many tags with varying frequencies may be assigned to the same item as shown in figura 2.1.

This plot displays how people tag a specific bookmark. The horizontal axis measures how many different tags are used for describing a given web page, while the vertical axis measures the frequency of each tag (or the number of times each tag is used). There is a small subset of tags that are strongly dominant and commonly acceptable with a high frequency. This means that a tag has been annotated several times by different users, which makes its weight increase and finally get more significant than others (even among those assigned to the same information item). Conversely, a wide range of different low frequency tags may be available as a long tail at the right end of the curve, where there is a minority of people who call the object with a personal terminology that is rarely re-used.

In fact, the form of tagging tends to stabilize over time because people usually choose to use the tags in three ways:

1. Imitation, users are easily affected by the tags that were previously applied by others to the same page;
2. Habit, users re-use tags that they have already used on other pages respect to their background and culture;
3. Recommendation, users choose tags that are suggested by a given interface.

Because of these patterns of tagging, after hundreds of people tagging the same bookmark a set of tags emerges as being capable to suitably describe the (generic) resource for any new user. Afterwards, almost no new tags are added by subsequent users, so that the tagging action tends to a stable distribution.

2.4 Limitation

2.4.1 Semantic Ambiguity

The social tagging system, based on multiple users and multiple words available for tagging, is in fact a semantic web application, which has the inherent vocabular confusion problem leading to a series of limitations and weaknesses. Although Golder and Huberman[3] have inferred that users could imitate the choices of tags applied by others, different tags used by different people for the same resource can still be present in the absence of a vocabulary control and without a standard to restrict user's tagging action. In fact, untrained people always use different form of words for their personal use, even one person himself also changes his own idea about tagging (and words available for tagging) over time. Therefore, tags have been noted to be inaccurate and leading to various ambiguities.

Polysemy

Polysemy is a word that has multiple different but related meanings. For example, 'bank' has a meaning as financial institution and also means the building where banking services are offered.

Homonymy

Homonym is a set of words with the same spelling and pronunciation, but with completely different meanings without any relations. For example 'Apple', is identified as product of Apple Inc. such as iPod, iTunes, Mac laptop and so on, but traditionally it just means the pomaceous fruit of the apple tree.

This is an important problem that can obfuscate the similarity among tags and resources. In other words, it can give a false impression that resources with similar tags also display similar content.

Acronyms

Acronyms represent another potential ambiguity similar to homonymy, where one form of tag has several meanings. For example, the web pages tagged by "PCC" on Delicious include the following items:

- PCC Home: Program for Cooperative Cataloging (PCC)
- pcc - pcc portable c compiler
- MyPCC Login -Portland Community College
- Press Complaints Commission >> Home Page
- Proof-Carrying Code

These three types of problems emerge because the tag-users apply the same tag in different ways. Hence, when users use the collaborative system to search informations, the same tag may return a result different from that expected. But it does not seem an important problem, because users can add related tags to restrain the query for future retrieval.

Synonymy

Synonyms arise when different words have a similar meaning. They represent a serious problem for tagging systems. Because of the inconsistency of the words used in tagging, a seeker may find it very difficult to ensure that all of the contents of retrieval were found. For example, if a resource can be indexed (say randomly) under the tags "man", "male" and "human", then a seeker may choose one tag to find resources, and once found out a correct term, he will stop searching without knowing that there are other useful resources with other tag synonyms.

2.4.2 Formative Drawbacks

Besides these semantic problems, the lack of clear tag structures greatly affects the use efficiency in applications.

singular vs. plural

The structure problem is that tags can be presented in different form, both singular and plural, which are all recognized as different tags. If we take a look at the Delicious, both "animal" and "animals" may be present at the same time. In this case, a search with only one word will not recover both of them, so we need an intelligent system capable of supporting these tags that differ only in terms of singular or plural.

Multiple word

Some collaborative systems such as Delicious allow users to record a single tag with several words together without spaces, such as "songforyou". With this method, users can use a single tag (for their own hierarchy category) containing more words, like "folksonomy/tag". But this kind of tag is only useful for its creator, not searchable by others.

Foreign Languages

An additional problem is tagging a document in foreign languages such as Chinese, Persian, Arabic, German, and French that are totally different words.

Nonsense words

One more problem exists in a word without meaning. Users can tag freely a resource with any word, there are examples like `ctt1`, `123456`, or any other words. Mistagging due to spelling errors is also one type of word without meanings.

Overall, these various unbounded vocabulary could result in tag redundancy, increasing volume of data and hindering navigation.

2.4.3 Limited Search

As for searching, the limitation problem is generated by word specificity, which reflects a cognitive aspect of users. Thus different users are concerned with and use different ranges of a specific vocabulary.

For example, a user wants to save a web page about how to make Italian pasta such as spaghetti, lasagna. If he does not know the classification of pasta, he would tag the page as food, pasta, or Italian. However, if he knows well the category, he would note: gnocchi, fusilli, ragù, and so on. Hence, if a page is tagged in a specific way such as gnocchi in the example, then people searching for food or pasta cannot find it.

So many people with different background, knowledge and experience working collaboratively with different specific tags, together with ambiguity and other drawbacks mentioned earlier (like unclear tags due to synonyms, singular and plural, spelling errors, personalized tags), and the absence of any mechanism to indicate hierarchical relationships between tags[11], all lead to the limited search problem.

2.4.4 Solution

As tagging systems are planar structures, users can not browse their contents by categorized guides. At the same time, vocabulary problems are almost constantly generated.

To solve these problems, one method is to use one or more external language resources, such as a lexical ontology like «WordNet», which contains English words with semantic relations. Also, Alireza in her study has suggested to use a dictionary or thesaurus as a controlled vocabulary to solve

semantic problems (not only about synonyms, but also including hypernyms and hyponyms)[10,14]. This controlled vocabulary is a separate tool, and can monitor misspellings, singular or plurality, while also providing a guidance for taggers and seekers when choosing the correct term for their annotation and retrieval. Finally, this also aids the system to improve the search results provided by the search engine.

But not everyone agrees with the need of a controlled vocabulary or thesaurus for tagging systems, because this method limits the selection of tags and adds the job of managing. Clustering algorithms for grouping similar tags in any form of expression can also solve this vocabulary problem arising in collaborative tagging systems. In particular, this latter solution is more acceptable and recommended by the majority of specialists. By means of suitable clustering methods, the system can classify all tags automatically without human experts assigning documents to classes. This is the method that I want to detail in the following chapter.

Chapter 3

Preliminary of Clustering

3.1 Scopes of Clustering

3.1.1 Query Handling

Collaborative tagging systems, where contents are generally associated with tags aim to improve the quality and efficiency of information searching and retrieval by the system users. Therefore, query handling is the ultimate service by the system for the users. Also, some of the users only submit queries without tagging; they can freely explore the application.

When a user wants to search for informations, the collaborative tagging system allows multiple resources to be queried for, which works like the query mechanism in a search engine. But the difference with respect to the traditional search engine is that when a user enters a number of keywords in a search engine he then expects to search on via the engine itself, where this latter uses an algorithm to label the documents as textual data mining and then displays the relevant resources to the user[19]. With the tagging system, resources are tagged by users, by social intelligence, and resources are similar only if they have been tagged by many user in similar way. This means that the similarity between resources attains through similarity between tags[24],

and when it comes to dealing with the information retrieval, a tagging system does not need to assign keywords to the resources, but follows and filters users' tags are already connected to resources. When a user selects a tag or puts a keyword in a search engine, the system then takes the tag and converts it into the automatic output of the clustering algorithm, dealing with it as a regular query, after matching the tag with related tags in the tag space, then finally displaying the relevant results to the user. In addition, some users even select a result of a query as the next query itself for further information seeking. In fact, it is the tag which constitutes the query for searching resources.

3.1.2 Organization of Resource Collection

Information retrieval concerns how to find resources relevant to a query, and it is generally hard to handle queries within a huge number of uncategorized resources. Thus, clustering makes sense of uncategorized resources and offers a browsing interface for the collection of resources.

3.1.3 Overcoming Ambiguities

A collaborative tagging system based on users' words is a semantic domain of co-occurring tags. As already mentioned, tags through which users communicate about available resources are chosen freely by users. They could be expressed differently by taggers and searchers in various forms with different or similar sense, which leads to a series of ambiguity and redundancy problems.

Hence, the tagging systems need clustering analysis and, more generally, data mining techniques to combat noise and redundant tags as well as to overcome redundancy and ambiguity problems[5]. By means of such techniques, queries can be handled through combining similar data points into clusters, which is more robust than searching by means of a single tag. Since a tag

cluster contains tags that are similar (given users' tagging behaviour, see below), the ambiguous meaning of a single tag used alone becomes relatively less important within a tag cluster, and the effect of ambiguity can also be remedied.

3.1.4 Recommendation

Collaborative tagging systems commonly possess a large number of available documents or resources, which are factually added every day. With the help of clustering, the tagging system can provide an application of recommendation and in particular both for tag recommendation as well as for personalized resource recommendation.

Tag recommendation refers to suggesting the most popular and useful tags to users when they want to add tags to resources. It is based on the historical informations on what others have tagged[18].

Resource recommendation concerns how to suggest resources to users when they select a tag for exploration. Through resource recommendation, the system suggests more similar resources to users based on tag clustering results, and possibly also relying on the users' profile to improve the quality of recommendation.

3.2 Clustering

Data clustering is the process of partitioning a data set into subsets or data clusters or blocks, whose members are as similar as possible between themselves, while being as "dissimilar" as possible to members of other clusters. In tagging systems, the goal of clustering is to minimize distances between tags in each cluster and maximize distances between clusters, according to the distance and/or similarity measure adopted.

In collaborative tagging systems, the set being partitioned is a collection of tags or tag set T , and the partition P is a set of clusters or subsets of T or blocks, while K is the generic number of desired clusters. The notation shall be:

- $T = \{t_1, \dots, t_I\} = \{t_i : i = 1, \dots, I\}$
- $P = \{P_1, \dots, P_K\}$ are K non-empty tag subsets or clusters, such that:
 1. $\emptyset \neq P_k \subseteq T$ for all $k = 1, \dots, K$
 2. $P_k \cap P_{k'} = \emptyset$, as $1 \leq k \leq k' \leq K$
 3. $P_1 \cup \dots \cup P_K = T$

The Process of Clustering

It seems useful to (ideally) divide the process of clustering into the three stages outlined below:

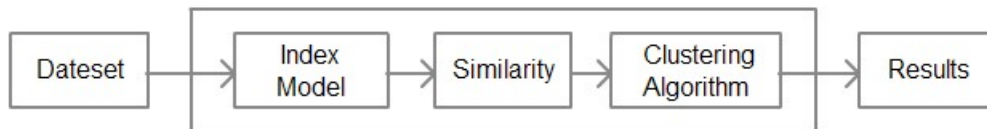


Figure 3.1: process of clustering

- Index model. First we need to consider how to store the dataset within the system. In other words, the issue is what kind of data structure to use in order to ensure the efficient storage of dynamic data and facilitate calculations of similarity between data. The index model is designed to achieve this target. It needs to consider how to create efficiency, namely space efficiency, retrieval efficiency, dynamic efficiency and semantic capabilities. Different similarity measures often require different index models, such as the Extended Boolean model, the vector space model, or probability theory in general.

- Similarity. A notion (or, more precisely, a measure) of similarity is needed to automatically clustering together the data with similar characteristics. Based on this similarity measure, clusters can be coherent internally as well as different from each other. It is central to all clustering analyses.
- Implementation of a clustering algorithm. Once available a similarity measure, the clustering algorithm has to partition the data so to reflect the similarities between them.

3.3 Similarity

The similarity method is the key input of most clustering algorithm. In order to properly partition tags into clusters, both a distance measure and a similarity measure can be used to reflect tags relations. With a distance measure, data within the same cluster must be close to each other, while with a similarity measure co-occured tags should have large association value in one cluster.

However, a good choice of the appropriate metric is very important, because two data points (or even two clusters) may be close to each other according to one similarity measure, but far away according to another measure. This influences the shape of clustering. Thus, different distance or similarity measures may be suitable for different clustering problems.

3.3.1 Distance

Dissimilarity can be measured by a distance measure, which is a quantitative variable. Given a generic data set $N = \{1, \dots, n\}$, $(a, b) \in N \times N$, the distance function is $dis : N \times N \rightarrow [0, 1]$, where [22,24]

- $dis(a, b) \geq 0$, the distance is always non-negative;

- $dis(a, a) = 0$, the distance from a data point to itself is zero;
- $dis(a, b) = dis(b, a)$, the distance is symmetric;
- $dis(a, b) \leq dis(a, c) + dis(c, b)$, the distance measure satisfies the triangle inequality;

The normalized dissimilarity matrix $Dis \in [0, 1]^{N \times N}$, where $Dis_{ab} = dis(a, b)$, $a, b \in N$. This is easy to transform into the similarity matrix, as $S_{ab} = 1 - Dis_{ab}$.

For n-dimensional data, a popular measure is the Minkowski Metric[28], as $dis(a, b) = \left(\sum_{k=1}^n |a_k - b_k|^p \right)^{\frac{1}{p}}$;

- If $p=1$, it is the Manhattan distance.
- If $p=2$, it is the Euclidean distance, which is one of the most common used distance measures (examining the root of square differences of pairs of data). It is defined as:

$$\|a - b\|_2 = \sqrt{\sum_{k=1}^n (a_k - b_k)^2}$$

- Case $p \rightarrow \infty$ is also useful in certain applications.

3.3.2 Similarity

The similarity function is $s : N \times N \rightarrow [0, 1]$ where $s(a, b) = 1$ is the similarity of a data with itself, also $s(a, b) = s(b, a)$ (symmetry). $Sim_{ab} = s(a, b)$ quantifies the similarity between any pair of data points a and b . It is a method based on probability, which takes a value between zero and one, where one means that the two data points are identical, and zero indicates that the data points are totally different.

Various notions of distance between subsets $A, B \subseteq N$ lead to different calculations using coefficients such as Cosine coefficient, Jaccard coefficient, or other similarity measures related to the Jaccard index (like Dice coefficient and Overlap coefficient)[11]:

- Matching

$$|A \cap B|$$

- Cosine coefficient

$$\frac{|A \cap B|}{\sqrt{|A| \times |B|}}$$

- Jaccard coefficient

$$\frac{|A \cap B|}{|A \cup B|}$$

- Dice coefficient

$$\frac{2|A \cap B|}{|A| + |B|}$$

- Overlap coefficient

$$\frac{|A \cap B|}{\min(|A|, |B|)}$$

Except for the matching method, most of these similarity coefficients count the number of elements appearing in the intersection and next divide it by a suitable term reflecting that the two subsets are independent. We can choose one of them as an example for dataing how clustering algorithms function.

3.4 Types of Clustering

3.4.1 Hierarchical Clustering

Hierarchical clustering is an algorithm that generates a sequence of clusters with a hierachical structure, where the root cluster is the coarsest partition consisting of a unique cluster or block that contains all of data points. Basing on clusters that have been previously established to find successive ones, it is usually divided into two classes, either agglomerative("bottom-up") or divisive("top-down") depening on the method [11,25]:

- Agglomerative algorithms start with the data space, where each element is considered as a separate cluster, and then the nearest (according to the chosen similarity measure) clusters are merged continuously into successively larger clusters, until achieving the predefined number of clusters;
- Divisive algorithms start with the whole set as a single cluster and then divide clusters into successively smaller ones, until achieving the predetermined number of clusters.

3.4.2 Partitional Clustering

Partitional clustering is a divisive hierarchical method, including k-means clustering, quality threshold clustering and graph methods as special types. K-means clustering is one of the simplest clustering algorithms, based on an input specifying the number of cluster K . It first randomly generates K cluster centers (usually in a suitable Euclidean Space), and then assigns all data points to the nearest cluster center. Given this initial allocation, the algorithm next iteratively calculates the new centers as the average of all data points within each cluster, and repeats the allocation of data until the center no longer moves and becomes stable.

3.4.3 Spectral Clustering

Spectral clustering techniques[11] make use of the similarity matrix, quantifying similarity within data pairs, and aiming to perform a dimensionality reduction.

3.4.4 Hard Clustering and Fuzzy Clustering

In hard clustering, the aim is to group data points into a partition, so that clusters are blocks. Every data point belongs to exactly one cluster. In fuzzy

clustering, every data point may belong to more than one cluster, with a membership ranging in $[0,1]$.

Chapter 4

Clustering Methods In Collaborative Tagging Systems

4.1 Structure of Tagging Systems

Before discussing clustering methods, the (generic) collaborative tagging system introduced in previous chapters has to be turned into a conceptual model that can be used by clustering algorithms.

A collaborative tagging system can be represented as a tuple $D = (T, U, R, \tau)$ where T, U, R are three main types of entities and τ is the action tagging:

- The tag set $T = \{t_i, i = 1, \dots, I\}$;
- The set $U = \{u_j, j = 1, \dots, J\}$ of users in the system;
- The set $R = \{r_m, m = 1, \dots, M\}$ of tagged resources (like websites in delicious);
- The generic tagging action $\tau \subseteq U \times T \times R$, $\tau_{ijm} = \{(t_i, u_j, r_m) : t_i \in T, u_j \in U, r_m \in R\}$ which contains the tagging information, namely that user u_j has associated resource r_m with tag t_i .

Each entity is in a separate data space, hence there is a tag space, a user space and a resource space. The set of nodes below represent the elements in each space, while links between them formalize tagging actions.

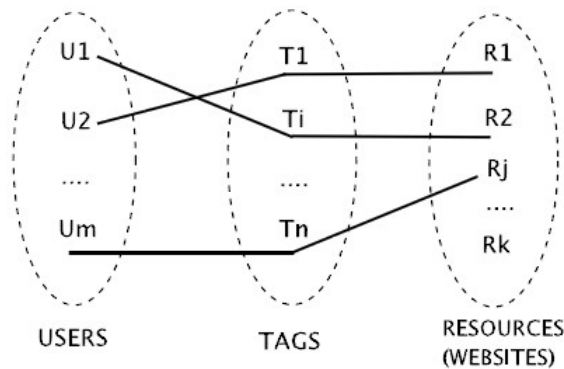


Figure 4.1: structure of tagging system

In fact a resource can be associated with one or more tags, and users can also associate any single resource with different tags. Tags serve as the nexus between users and resources, as shown in figure 4.1. When a user wants to browse a collaborative tagging system, he can freely navigate through three dimensions: tags, resources or other users. The management of resources does not depend on the digital nature of these latter, but relies only on user's tags. Thus, what is essential of each resource for each user can be captured by its associated tags. Hereafter, clustering methods automatically producing tag partitions for collaborative tagging systems are discussed.

4.2 Hierarchical Clustering

Hierarchical agglomerative clustering is a traditional clustering method, fruitfully applicable to tagging system. It ranges over a series of clusters P_1, \dots, P_I , from P_I consisting of i elements to P_1 containing all i objects together. This algorithm has been suggested for collaborative tagging systems as a tool for improving recommendations .

In a system $D = (T, U, R, \tau)$ as we have denoted it above, there is a tag set $T = \{t_i, i = 1, \dots, I\}$ and a resources set $R = \{r_m, m = 1, \dots, M\}$. Basing on the vector space model, each of them is modeled as a vector, hence a tag is a vector over the set of resources. We define the tag frequency as:

$$tf(t_i, r_m) := |\{(t_i, u_j, r_m) \in \tau : u_j \in U\}|$$

which is the number of times a tag $t_i, 1 \leq i \leq I$ has been associated with resource $r_m, 1 \leq m \leq M$.

The similarity between every pair of tags is calculated according to Cosine similarity[5], that is,

$$sim(t_i, t_j) = \frac{\sum_{r_m \in R} tf(t_i, r_m) \cdot tf(t_j, r_m)}{\sqrt{\sum_{r_m \in R} tf(t_i, r_m)^2} \cdot \sqrt{\sum_{r_m \in R} tf(t_j, r_m)^2}}$$

This yields similarity matrix $Sim \in \mathbb{R}^{I \times I}$, where $Sim_{ij} = sim(t_i, t_j)$.

The distance or similarity between different clusters also plays a crucial role. It can be measured in different ways, such as single-linkage, complete-linkage, average-linkage. Each method leads to a different clustering result. In the single-linkage case for example, the distance between two clusters is the shortest distance between a tag of one cluster and a tag of the other cluster[28]. The distance function between two clusters $c_k, c_{k'}$ is $dis(c_k, c_{k'}) = \min dis(t_i, t_j) = \max sim(t_i, t_j)$, where $t_i \in c_k, t_j \in c_{k'}$ and $c_k, c_{k'} \subset T, c_k \cap c_{k'} = \emptyset$.

For collaborative tagging systems, we set:

- (1) a parameter g controlling the granularity of clustering;
- (2) a similarity threshold which is initially 1, and next gradually it is reduced to 0; for every pair of clusters if their similarity meet the current threshold, they are joined together;

(3) a division coefficient serves to decide at which level of the hierarchy clusters should be split into individual elements[5]. This coefficient is crucial for determining the final number of clusters.

Relying on the basic process of hierarchical clustering(S.C. Johnson in 1976)[28], the associated algorithm may be outlined as follows:

Table 4.1: Algorithm of hierarchical agglomerative clustering

Input:

A set of tags T , similarity matrix Sim , parameter g , a division coefficient DC .

output:

A set of tag clusters C .

Method:

- Start by assigning each tag to a cluster $c_i = t_i, (i = 1, \dots, I)$. So that if there are I tags, then there are I clusters $C = \{c_1, \dots, c_I\}$, (that is every cluster contains just one tag).
- Set the iteration index L to 0.
- Supposing the similarity threshold $ST = 1 - gL$, where g decides the granularity of clusters (see above).
- **Repeat:**
 - **For** each pair of clusters $(c_k, c_{k'})$ **do**
 1. Set iteration index $L = L + 1$, update ST and get the current similarity threshold;
 2. Find a most similar pair of clusters $(c_k, c_{k'})$, and merge them into a single cluster; more precisely if $sim(c_k, c_{k'}) \geq ST$, let $c_l = c_k \cup c_{k'}$ so that most similar clusters are iteratively joined together;
 3. Update the cluster set by deleting $c_k, c_{k'}$ from C and inserting c_l into C ;
 4. Update the similarity matrix by computing the distances between new clusters based on single-linkage algorithm.
 - * This is a process based on the similarity matrix: firstly, the two closest or most similar tags are combined together; secondly the similarity between the new cluster and the old clusters is computed, and next the merging step starts again.
 - **End for**
- **Until** all tags are merged into a single hierarchical cluster that can be represented with a dendrogram like a tree diagram(see fig. 4.2).
- Split the tree structure into clusters according to the division coefficient DC .

As hierarchical clustering does not require a predeterminate number of clusters, it is only necessary to cut branches out of the tree at a level of similarity established by the division coefficient, thereby getting the disjoint clusters just as in flat clustering.

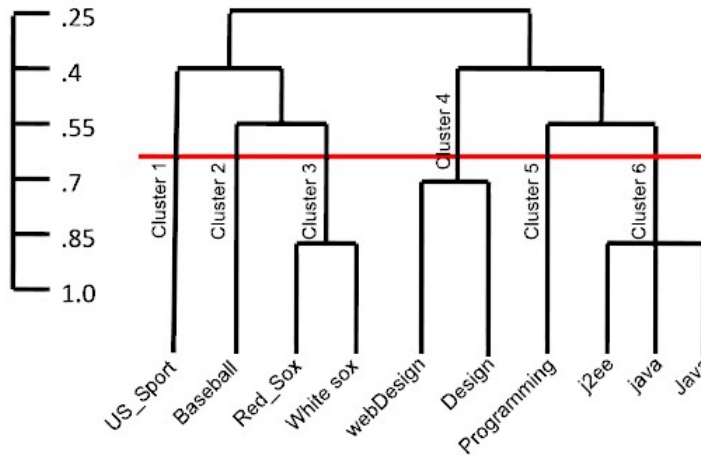


Figura 4.2: Hierarchical agglomerative clustering

In this algorithm, parameter g is used to control the granularity of agglomerative clustering: the smaller the value of g , the more hierarchical levels the algorithm will generate. As shown above, with $g = 0.15$ the similarity threshold decreases by 0.15 at every step. After five steps, all data are combined in one cluster. By choosing a suitable value of g , the process of aggregation becomes slow enough so to capture the relationships between tags in each cluster[29]. Otherwise, some interdependencies may be lost.

In addition, the division coefficient determines the similarity threshold for cutting the hierarchical tree, and any cluster below the level is considered as an independent cluster. For example, in figure4.2 above, the division coefficient is such that tags have been divided into six clusters. If we take a high value for division coefficient (close to one), then we get many small

clusters containing very similar tags. In contrast, a low value for the division coefficient leads to large clusters with low internal similarity.

However, hierarchical clustering displays quadratic time complexity, and is considered to be an high quality clustering approach.

4.3 K-means Clustering

K-means clustering (MacQueen 1967) is one of the simplest, widely used and unsupervised clustering approach, with the aim to partition a data set into a predetermined number K of blocks. It is an efficient and highly scalable clustering method, fruitfully applicable to collaborative tagging systems.

We use K-means clustering to partition n tags into k clusters by minimizing the distance within each cluster, while also maximizing the distance between clusters. First we need to assign K centroids, one for each cluster. They should be quite far away from each other in the relative metric space. Then, each cluster centroid is iteratively recalculated, and next each tag is reassigned to the closest cluster, until no tags can be reassigned. Indeed, assigning tags to the closest cluster is achieved by minimizing the distance between tags and their cluster centroids. After assigning all tags to the K clusters, at each iteration the centroids need to be recalculated as the average of the tag vectors in each cluster.

For K-means clustering, we also use vector spaces to represent the three entities of collaborative tagging systems, as each of them is modeled as a vector. Consider the same tag frequency tf and the same similarity measure $sim(t_i, t_j)$ between tags introduced above for hierarchical clustering.

Table 4.2: Algorithm of K-means clustering

Input:

A set of tags $T = \{t_i, i = 1, \dots, I\}$, the number of clusters K

output:

A set of tag clusters $C = \{C_k, k = 1, \dots, K\}$.

Method:

- Initialize the number K , $k = 1$, $\mu_1 = t_1$, $C_1 = \{\mu_1\}$
- assign K tags as initial centroid μ_j to each cluster, as dissimilar as possible from each other.
 - **For** $k = 2$ to K **do**:
 - * $\mu_k = t_i : \min_{i=1}^N \text{sim}(t_i, \mu_{k-1})$;
 - * $C_k = \{\mu_k\}$;
 - **End for**
- **Repeat**:
 - Reassign all tags to the closest clusters based on the similarity measure, where each tag is nearest to its centroid.
 - * **For** $i = 1$ to I **do**
 - Put t_i into each cluster C_k , where t_i satisfies $\max_{k=1}^K \text{sim}(t_i, \mu_k)$
 - * **End for**
 - Iteratively update centroid in each cluster
 - * **for** k from 1 to K **do**:
 - Recalculate and replace each centroid μ_k based on the tags in cluster C_k , as $\mu_k = \frac{1}{|C_k|} \sum_{t_i \in C_k} t_i$
 - * **End for**
- **Until** the centroid of each cluster no longer changes.

The choice of the number K of desired clusters is very important, of course,

for this algorithm. In fact, a great deal of attention has been paid to finding the optimal number of clusters for given data set. Evidently, an inappropriate choice may lead to very poor results.

The time complexity of this K-means algorithm is linear in the number of tags. It is an efficient method in terms of run time, better than the hierarchical clustering algorithm.

According to the comparison of between clustering algorithms performed by Shepitsen et al.[5], the K-means clustering algorithm has a drawback, in that it cannot efficiently deal with outliers neither isolate irrelevant tags. The issue arises because all tags have to be assigned to K clusters. Hence, some of the clusters may contain more than one topic area, thereby confusing the aggregate meaning of the cluster. However, in hierarchical clustering strongly related tags are aggregated together rapidly, while less related tags are considered afterwards.

4.4 Relational structure and graphs

A graph structure is often used to model relations between data points. Traditionally, these latter are always dealt with as feature vectors, but in more recent times graph theory seems to be more useful for modeling pairwise relations between objects such as network data. Also, data points in the form of vectors can be transformed into a graph which is easier to deal with.

A graph is a pair $G = (V, E)$, where V is a set of vertices, and E represents the edges of G , $E \subseteq \{A \subseteq V : |A| = 2\}$.

A subgraph $G' = (V', E')$ of graph G satisfies $V' \subseteq V$ and $E' \subseteq E$.

(Any such a simple graph as an associated graph partition problem which is NP—complete.)

4.4.1 Preparing Dataset

By means of graph theory, we use an undirected graph to represent the relational structure between tags which is constructed through tag co-occurrences. The relation structure takes the form of a simple graph $G(T, E)$, in which T is the collection of tags and E is a set of edges or links between tags.

For example, a tag relation graph taken from delicious is

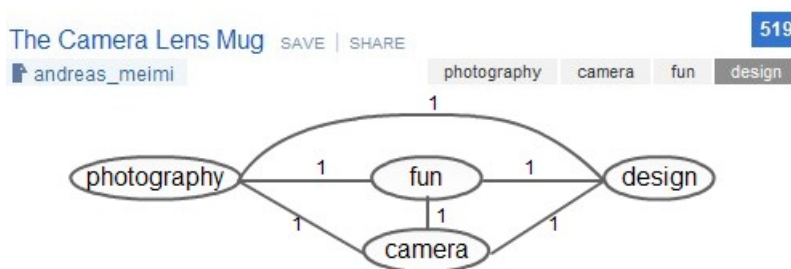


Figure 4.3: tag-relations

A certain bookmark is associated with the following tags: photograph, camera, fun and design. Then the edge weight of each pair of tags, like (design, fun) and (design, camera), get one count as co-tags with the aim to build a tag relation graph, where each tag relates to all the other tags, through a positive (possibly zero) weight obtained by counting co-occurrences, as shown in figure 4.3.

Again if we sum up all these small tag relations into unique overall graph, then this latter will be too big for tag retrieval.

4.4.2 Strongly Related Tags

Since tagging systems allow for uncontrolled vocabulary, the power law distribution of tagging discussed in previous sections, shows that just a small subset of tags are strongly dominant for any given resource, which makes their frequency higher than the other little used tags. According to the

power law distribution, we can also infer that a small subset of tags will highly co-occur, and such a subset thus contains strongly related tags. The long tail of tags for personal use is not reinforced by the majority of users, but when considering co-tags they cause a lot of noise (in the form of extra edges with low edge weight, reducing the quality of clustering[17]). If there are 100 tags in tag space, then the graph has $\binom{100}{2} = 4950$ edges, one for every pair of tags. Taking into account the noise relationship, only a top fraction constitutes strongly related ones.

In order to compress the tag space and find out what tags are strongly related, we can use the similarity method by counting co-occurrences. Then, we need an acceptable tunable threshold above which the co-occurring tags are considered to be strongly related and thereby maintained. Conversely, the weight of weakly related tags shall be less than the threshold, and thus pruned.

4.4.3 Similarity

For tagging systems, the input of clustering algorithms is $D = (T, U, R, \tau)$, containing tags $T = \{t_i, i = 1, \dots, I\}$, users $U = \{u_j, j = 1, \dots, J\}$, resources $R = \{r_m, k = 1, \dots, M\}$ and also the tagging behaviour τ .

Now let the tagging τ be a 3D tensor: $\tau \in \mathbb{R}^{I \times J \times M}$, where $\tau_{ijm} = 1$ means a tag t_i has been tagged by a user u_j to a resource r_m , and $\tau_{ijm} = 0$ otherwise.

For computing the similarity between pairs of tags, we take a matrix $H \in \mathbb{R}^{I \times M}$, where $H = \vee_j \tau_{ijm}$ represents the tagging information from all users who used t_i for r_m , while \vee_j means the logical OR. Its rows correspond to tags, while columns correspond to tagged resources. In order to calculate the similarity of co-occured tags, we take the (i, j) -entry of H for the simple matching or Jaccard coefficient method.

Matching

Matching, also known as «strength of connection», is the easiest method. It relies upon the count of the number of resources associated with both any two tags t_i, t_j , and how many users made such an association. The similarity between pairs of tags[19] is:

$$||H_i \wedge H_j||_1$$

where \wedge is the logical AND and $||\cdot||_1$ stands for the L_1 norm (of a boolean vector[6]). With this method, we can find the significant co-tags by comparing their frequency count with the predetermined threshold, and cutoff weakly related tags to compress the tagspace.

By counting the number of times a pair of tags co-occured, the tags belonging to any single cluster should be highly connected.

Jaccard coefficient

Jaccard similarity coefficient, also known as the Jaccard index, is another method used for comparing the similarity of tags. The Jaccard similarity is defined as:

$$sim_{ij} = \frac{||H_i \wedge H_j||_1}{||H_i \vee H_j||_1}$$

where $||H_i \wedge H_j||_1$ as mentioned previously is the number of times two tags are annotated together for the same resources, and $||H_i \vee H_j||_1$ is the union of resources that contain any one of two tags.

4.4.4 Weighted Graph Creation

A preliminary step for any graph partitioning algorithm is determining a suitable (relational) graph. How to create a weighted graph is now briefly outlined:

1. Calculate the similarity between pairs of tags;
2. Determine a threshold;
3. If $sim_{ij} \geq threshold$, so that the similarity is larger than the chosen threshold, maintain the edge between the two tags, otherwise delete the edge;
4. take sim_{ij} as the edge weight for any surviving edge;

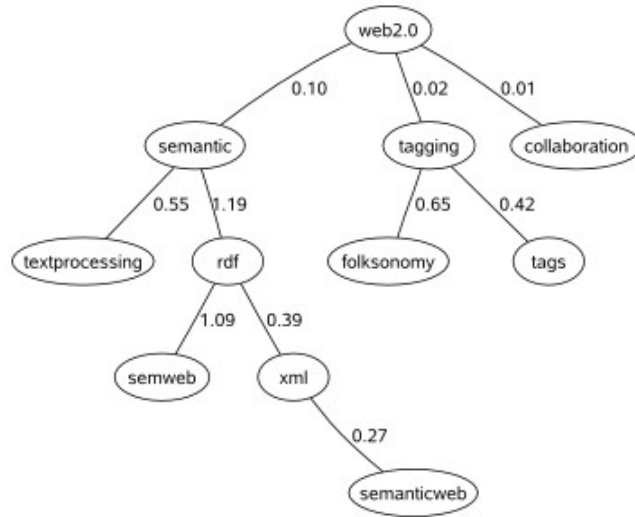


Figure 4.4: weighted graph

By dealing with all tags in the tag space, we can get a simple undirected tag-tag weighted graph $G(V, E, W)$ [6], where:

- $V = \{v_i, i = 1, \dots, I\}$, is the set of vertices in the graph, each vertex v_i in the graph corresponds to a tag t_i . For shorthand notation, we sometimes take i for the set of vertices $\{i | v_i \in V\}$;
- $E \subseteq V \times V, \{v_i, v_j\} \in E$, is a set of edges linking strongly related vertices v_i and v_j .

- $W = (w_{i,j})_{i,j=1,\dots,I} \in \mathbb{R}^{I \times I}$ is a weighted adjacency matrix, which quantifies the similarity between any two vertices, while I is the number of tags. As G is an undirected graph, this adjacency matrix is symmetric, with $w_{i,j} = w_{j,i} \geq 0$ corresponding to the the weight on edge between vertices v_i and v_j . When $w_{ij} = 0$, there is not an edge between v_i and v_j , they are not connected.

4.4.5 A Clustering Algorithm

By means of the similarity weights, we now have a big undirected graph, and the aim is to partition its vertex set. In particular, the similarity across different blocks has to be low (the edges between blocks have low weight), but there has to be high similarity within blocks.

4.4.5.1 Spectral clustering

Spectral clustering a graph partition method firstly suggested by Donath & Hoffman 1972. It is an algorithm with high computational performance and relevant dimensionality reduction.

The main tool of spectral clustering is the graph Laplacian matrix $\mathcal{L} = [l(i,j)]_{I \times I}$. The unnormalized Laplacian is defined as[11,20,22]:

$$\mathcal{L} = D - W$$

which is the difference between the degree matrix and the adjacency matrix, the former being denoted $D \in \mathbb{R}^{I \times I}$. It is a diagonal matrix where all the degrees d_1, \dots, d_I of vertices are on the diagonal:

$$D = \begin{pmatrix} d_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & d_2 & 0 & \dots & 0 & 0 \\ 0 & 0 & d_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & d_{I-1} & 0 \\ 0 & 0 & 0 & \dots & 0 & d_I \end{pmatrix}$$

The degree of the vertex v_i is defined as[20]:

$$d_i = \sum_{j=1}^n w_{ij}$$

It is the sum of all weights of edges linking v_i to some other vertex.

The unnormalized Laplacian matrix can be defined as follows:

- $l(i, j) = d_i$, if $i = j$;
- $l(i, j) = -w_{ij}$, if $i \neq j$, $\{v_i, v_j\} \in E$;
- $l(i, j) = 0$ otherwise.

Matrix \mathcal{L} [11,20,26] satisfies the following properties:

- It has I eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_I$, where each eigenvalue is the number λ satisfying $\mathcal{L}x = \lambda x$ for a non-zero eigenvector x ;
- \mathcal{L} is symmetric and positive-semidefinite, that is, $x^T \mathcal{L}x \geq 0$, thus all eigenvalues of \mathcal{L} are real-valued and non-negative, as $\lambda_i \geq 0, 1 \leq i \leq I$;
- Let $\mathbf{1}$ be the constant vector of all ones: $\mathbf{1} = [1, 1, \dots, 1]^T$. If $\mathcal{L}\mathbf{1} = 0$, then we can say that 0 is the smallest eigenvalue of \mathcal{L} and $\mathbf{1}$ is the corresponding eigenvector;

- The multiplicity k of the eigenvalue 0 of \mathcal{L} corresponds to the number of connected components P_1, \dots, P_k in the graph, (or blocks of the partition of the vertex set), while the eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbf{1}_{P_1}, \dots, \mathbf{1}_{P_k}$ of components;

By computing the eigenvectors corresponding to the eigenvalues of Laplacian matrix \mathcal{L} and comparing this latter with the weighted matrix W , it can be observed that both have a block diagonal form:

$$\mathcal{L} = \begin{pmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_k \end{pmatrix}$$

where each of the block L_i corresponds to the i th connected subgraph.

Generally, an unnormalized spectral clustering can be described through the as following steps[20]:

Tabella 4.3: unnormalized spectral clustering algorithm

Input:

The similarity matrix Sim , the number of clusters K

output:

A set of data clusters $C = \{C_k, k = 1, \dots, K\}$

Method:

- Create a weighted graph corresponding to similarity matrix;
- Set weighted matrix W and degree matrix D ;
- Compute the unnormalized Laplacian \mathcal{L} ;
- Compute the first K eigenvectors x_1, \dots, x_K of \mathcal{L} ;
- Take the x_1, \dots, x_K as columns and generate an matrix $X \in \mathbb{R}^{I \times K}$;
- For $i = 1, \dots, I$, let the i th row in X be the row vector $y_i \in \mathbb{R}^K$;
- Cluster the data points $(y_i)_{i=1, \dots, I}$ into K clusters with K-means clustering algorithm or other vector-based clustering algorithm.

4.4.5.2 Normalized Cut

One technique of spectral clustering is the Normalized Cuts algorithm with the aim to find the small cut of a graph, thus the minimal connections between subgraphs. It is a spectral bisection method that iteratively divides vertices into two clusters. In 1973 Fiedler has pointed out that the bipartition of a graph is decided by the eigenvector of the second smallest eigenvalue of the graph Laplacian. We can use this eigenvector to partition a graph.

Based on the weighted graph, a set of vertices $V = \{v_1, \dots, v_I\}$ can be divided into two parts P and P' , $P \cup P' = V$, $P \cap P' = \emptyset$, where P is a subset of vertices ($P \subset V$) with complement $V \setminus P = P'$. Let $i \in P$ and $j \in P'$.

A partition of the vertices in two disjoint subsets is defined to be a cut:

$$cut(P, P') = \sum_{i \in P, j \in P'} w_{ij}$$

It obtains by removing the sum of weights associated with edges between two parts. In order to realize the optimal bipartition of a graph, we need to minimize the *cut* value, which means finding the minimal total weights of edges connecting different partitions and then divide them. This is based on what Shi and Malik[11,23] have termed a normalized cut(Ncut) :

$$Ncut(P, P') = \frac{cut(P, P')}{d_P} + \frac{cut(P, P')}{d_{P'}}$$

where $d_P = \sum_{i \in P} d_i = \sum_{i \in P, j \in V} w_{ij}$ is the sum of weights of all edges linking vertices in the graph to vertices in P , while $Ncut(P, P')$ measures the similarity between different blocks of the partition. Thus, the size of $Ncut$ is measured by the sum of weights of edges. This enables to avoid partitioning out just small sets of isolated nodes in the graph, displaying small cut value.

Let x be the cluster indicator vector defined by[20,23,26]:

$$x_i = \begin{cases} \sqrt{d_{P'}/d_P d} & \text{if } v_i \in P \\ -\sqrt{d_P/d_{P'} d} & \text{if } v_i \in P' \end{cases}$$

where $d = \sum_{i \in V} d_i$.

According to the normalized cut of Shi and Malik[23], after some manipulations, the Ncut can be represent as :

$$\frac{x^T \mathcal{L} x}{x^T \mathcal{D} x}$$

with $x^T \mathcal{D} \mathbf{1} = 0$ and $x^T \mathcal{D} x = 1$.

By considering a relaxation of the problem, which means taking real (rather than Boolean) values of x , we can solve the minimize Ncut problem by solving the generalized eigenvalue system[23]:

$$\mathcal{L}x = \lambda Dx$$

where x is the eigenvector corresponding to the smallest eigenvalue λ of \mathcal{L} .

When using the Ncut algorithm with bisection of a graph for tagging systems, we can solve the partition problem by computing the graph Laplacian \mathcal{L} , and get the eigenvector x corresponding to the second smallest eigenvalue of \mathcal{L} .

By relaxing the cluster indicator vectors allowing for real-values, the process of bisecting a graph can be done by using the sign of x according to following function:

$$\begin{cases} v_i \in P & \text{if } x \geq 0 \\ v_i \in P' & \text{if } x < 0 \end{cases}$$

and then iteratively repartition the graph into subgraphs, relying on the function $Ncut$. Once bisected a graph into subgraphs, recalculate the normalized cut value between two partitions and decide if the subgraph needs further cutting. We set a threshold of Ncut value, and if $Ncut < threshold$, then go for further bisectioning. Otherwise, stop and exit the bisection algorithm.

This can be summarized as follows:

Tabella 4.4: Normalized cut algorithm

Input:

The similarity matrix Sim , the number of clusters K

output:

A set of data clusters $C = \{C_k, k = 1, \dots, K\}$

Method:

- Create a weighted graph corresponding to similarity matrix;
- Set weighted matrix W and degree matrix D ;
- Compute the unnormalized Laplacian \mathcal{L} ;
- Compute the eigenvectors and eigenvalues of $\mathcal{L}x = \lambda Dx$;
- Take a look at the eigenvector x^2 corresponding to the second smallest eigenvalue λ_2 ;
- Assign vertices to c_k , if $x_i^2 \geq 0$, otherwise if $x_i^2 < 0$, assign vertices to c_j
- Compute the normalized cut between c_k and $c_{k'}$, if $Ncut$ value is small, thus $Ncut < threshold$, repeat above steps.
- Until $Ncut > threshold$.

Now consider a simple example provided by Lee[32]. Suppose the similarity between tags is one when they are co-occured together, otherwise the similarity is zero. The similarity matrix is

$$Sim = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Based on the similarity of co-occured tags, we create the graph and let the weight matrix represents the similarity between tags $W = Sim$:

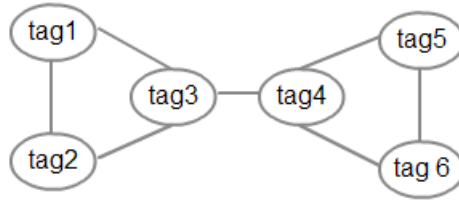


Figura 4.5: Related Graph

Then the Laplacian matrix $\mathcal{L} = D - W$ is computed as:

$$\mathcal{L} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

Then calculate the first three eigenvalues of \mathcal{L} as $\lambda_1 = 0$, $\lambda_2 = 0.439$, $\lambda_3 = 3$ etc., and also their corresponding eigenvectors as:

	x1	x2	x3
t1	0.4082	-0.4647	0.7071
t2	0.4082	-0.4647	-0.7071
t3	0.4082	-0.2610	0
t4	0.4082	0.2610	0
t5	0.4082	0.4647	0
t6	0.4082	0.4647	0

Figura 4.6: eigenvectors of \mathcal{L}

Take the eigenvector x^2 corresponding to the second smallest eigenvalue λ_2 . The tag set then can be cut into two clusters with the opposite sign of vector

x^2 , where $c_1 = \{t_1, t_2, t_3\}$ has $\text{sign } x^2 < 0$ and $c_2 = \{t_4, t_5, t_6\}$ has $\text{sign } x^2 > 0$. Calculate $Ncut(P, P') = \frac{cut(c_1, c_2)}{d_{c_1}} + \frac{cut(c_1, c_2)}{d_{c_2}} = \frac{1}{d_1+d_2+d_3} + \frac{1}{d_4+d_5+d_6} = \frac{2}{7}$. Then compare the Ncut value to threshold for deciding whether to further re-partition or not.

4.4.5.3 Modularity function

Whitey and Smyth[21] improve the Ncut value and introduce a modularity function Q to measure the quality of clustering in a graph, in order to compare which partition is optimal. Through this function the optimal number of clusters k can be automatically selected by maximizing Q , and the modularity function has been defined as:

$$Q(P_k) = \sum_{P=1}^k \left[\frac{A(V_P, V_P)}{A(V, V)} - \left(\frac{A(V_P, V)}{A(V, V)} \right)^2 \right]$$

where P_k is a partition of the vertices into k groups and $A(P, P') = \sum_{i \in P, j \in P'} w_{ij}$, while V_P is the set of vertices of partition P . On the other hand $A(V_P, V_P)$ is total weight of edges connecting within blocks (all these edges have both ends within the same block), $A(V_P, V)$ is the sum of weights of edges between vertices in the block and vertices in the whole graph, $A(V, V)$ is the total edge weight in the graph. Intuitively, $A(V_P, V_P)$ contributes to increasing modularity, while $A(V_P, V)$ gives a negative effect to modularity. Thus, the modularity Q can be informally considered as the deviation between the probability that both ends of a randomly selected edge in the graph fall within a block, and the probability that a randomly selected edge has its ends in two different blocks.

Newman[33] identified blocks of partitions as communities. He suggested to calculate the modularity function Q at every iteration of clustering, until there is no improvement. As shown in [33], if $Q = 0$, then the chosen partition P is no better than a random division. Conversely, if $Q = 1$, this is a strong community structure no edges connect nodes across clusters. In practice,

in real networks a value of Q greater than 0.3 indicates a high community structure.

Based on the spectral bisection method, the graph partition can combine the modularity with a recursive greedy clustering algorithm. Similarly to the Normalized cut algorithm, this method performs the following steps:

- Initially set $k = 1$ as the current number of clusters, thus one cluster contains all vertices in the graph. Set Q_1 corresponding to the unpartitioned graph.
- Compute the graph Laplacian \mathcal{L} based on weighted graph and also use the second smallest eigenvalue to bisect the graph into two subgraphs.
- At each iteration of bisection of graph, calculate the modularity value of these two subgraphs Q_2 .
- Compare Q_2 and Q_1 . If $Q_2 > Q_1$ accept the split, otherwise reject the partition and do not change the graph.
- Update k as the current number of clusters.
- Repeat the above steps until the modularity function Q does not further increase.

4.5 Fuzzy Clustering

We have discussed some hard clustering algorithms, through which data points are divided into distinct clusters or blocks of a partition. In fact, many works concerning collaborative tagging systems adopt with hard clustering methods. Conversely, in fuzzy clustering each data point has a degree of membership in existing clusters, and thus each data point can belong to more than one cluster[11]. In particular, fuzzy c-means algorithm is a very

popular one. However, this method is rarely conceived for and adopted in tagging systems.

One example is provided by Han and Chen[35], who present a fuzzy clustering method for collaborative tagging systems. They combine the fuzzy c-means algorithm and a subtractive clustering algorithm together to deal with social tagging problems. Han and Yan[36] also present a fuzzy bi-clustering algorithm especially designed for dealing with the problem of social annotation, which is addressed by partitioning users and resources into subgroups.

Dong et al.[37,38] present a hierarchical clustering algorithm using fuzzy theory. In their algorithm, a fuzzy graph $G_f = (V, E, L)$ is generated, in which V is a set of vertices representing data points and E represents a fuzzy edge relation, L is the level of presence of each fuzzy edge in the graph, taking value from zero to one[22]. Considering a threshold l_t , one can then obtain different non-fuzzy graphs, each containing those edges with presence index greater than threshold: $l(i, j) \geq l_t$. Essentially, this is a cut graph of G_f (see above). In their clustering method, they first partition the data set into clusters with the similarity coefficient, then analyze the fuzzy degree among the clusters, based on which generate fuzzy graph. Using the cut graph for fuzzy graph, as the result of a clustering algorithm.

Although this fuzzy graph method is not used in current collaborative tagging systems, it is possible to realize it in future studies.

4.6 Time sensitiveness

Time sensitiveness (Begelman et al.[6]) means that once a tag clustering is available, it does not remain valid all the time. The tagging behavior of users changes over time, thus the tag clusters need to be updated periodically.

Ning et al.[31] also suggest a real time update algorithm based on spectral clustering in graph, through which the system can insert and delete data points and change similarity between current items.

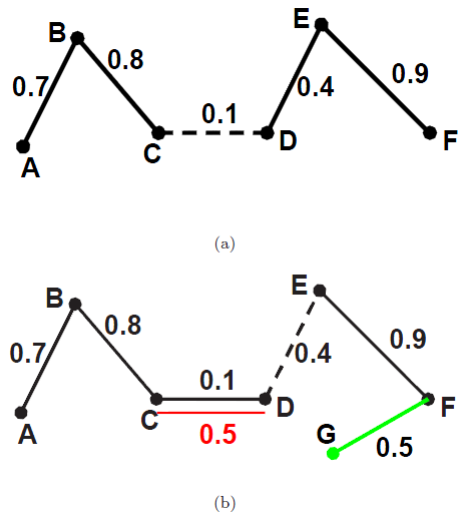


Figure 4.7: update data set

As is shown in figure 4.7, once update the data set, the relationship is changed form (a) to (b), in which a new strong related datum added in the graph, and the similarity between two vertices C and D is changed form 0.1 to 0.5. If we cut the graph edge when its similarity edge weight is small, thus we may cut CD in figure(a) with the weight 0.1, but we change to cut DE in figure (b) where its edge weight 0.4 is the smallest.

So Ning et al. introduce a dynamic framework using Laplacian matrix and a vector represent similarity change. Once a vector of changed simialrity is added, update the graph Laplacian, the degree matrices and also its corresponding eigenvalues and eigenvectors. Basing on the current eigenvalues to update the process of clustering.

Chapter 5

Recommendation In Collaborative Tagging Systems

Collaborative tagging systems determine the the content of resources depending on users' tags, thus the similarity between resources is affected by users' tagging. Once using the data mining techniques to automatically clustering tags, it is available providing different services to users.

As we have noted previously, the collaborative tagging system comprise three entities—users, resources and tags. Basing on the clustering of tags, in which tags are disambiguated, the system is able to track more accurately the common interests of users by measuring users over tag clusters, and resources associated with tags clusters can also be captured for information retrieval. In this type of systems, tag clusters paly the role of a intermediary that connect uesrs and resources, though which similar resources can be aggregate based on similarity of tags, and also similar users with common interestes can find each other in a group. Gemmell et al.[5] has pointed out that tag clusters can also serve as intermediries between two users to identify like-minded individuals for the construction of social networks. In addition we can recommendate relevant resource to users.

5.1 Personalized Recommendation

One application of tag clustering is searching to achieve information retrieval and suggest relevant resources to users. In deed, information retrieval is the main application in collaborative tagging system. Users may share information resources collaboratively with tags and then they can seek useful resources based on a mass of users' tags, thus based on social intelligence. After clustering the tag set in groups, the tagging system can implement more effectively information retrieval.

Traditionally, web search engines need extracting keywords from the content of documents or web resources such as titles, headings,etc. Then these information of data are stored as index in database for future queries. When a search engine recieved a query from a user, it then analyzes the index and match to its web resources, then return the a list of top n related resources to the user.

For collaborative tagging system, tags created by users are stored as index data for use in queries. The system does not need to store large of information data about the resources, such as a part of the text in document. When a user search for informations in a tagging system like delicious, he or she can do it with a search query from a seach engine interface or just select a tag from the tag list, and then compare the query tag to resources and return a subset of document to users.

So we can assume that the user's search query is a specific tag. With this tag, the system can recommendate a set of resources. Then considering the users profile with the tag clusters, the system can re-ranking the resources and apply a personalized recommendation.

We have denoted the collaborative tagging system $D = (T, U, R, \tau)$ in the vector space model as in hierarchical clustering algorithm, each of them is modeled as a vector, like a tag is a vector over a set of resources, a resource is a vector over a set of tags, also a user is a vector over a set of tags. We have defined the tagging τ is $\tau_{ijm} = \{(t_i, u_j, r_m) : t_i \in T, u_j \in U, r_m \in R\}$ and the

tag frequency as: $tf(t_i, r_m) := |\{(t_i, u_j, r_m) \in \tau : u_j \in U\}|$ is the number of times the tags have been annotated to the resources.

The core of the search engine in tagging system is the ranking algorithm. Relying on the recommendation algorithm denoted by Shepitsen et al.[29], we describe the the process of recommendation as following phases.

The process of query handling start by computing the similarity between query q and resources r_k . This is the main task in resource recommendation. Taking a vector to represent a query, in which fill with zeros in the vector expect for the target query tag. we defined the similarity of a query tag and resources according to cosine similarity, as:

$$sim(q, r_m) = \frac{tf(q, r_m)}{\sqrt{\sum_{t_i \in T} tf(t_i, r_m)^2}}$$

Compare the query tag to every resource by computing the similarity $sim(q, r_m)$. Set a threshold to filter the resources with high similarity to the query as we need. Then as the output, we can get a subset of resources R' and all future phases are based on this subset.

In order to produce a personalized recommendation, we need to then match the obtained tag clusters $C = \{c_1, \dots, c_K\}$ against the user profile U , to obtain the users' interest. A feature of users navigation is the way that the user tagging a resource. If we know the type of tags that a user usually used, we could use these tags to recommendate resources. The tag produced on the resource is a good measure of user's interest. We defined it as:

$$uc(u_j, c_k) = \frac{|\tau_{ijk} : t_i \in c_k, r_{m'} \in R'|}{|\tau_{ijk} : t_i \in T, r_{m'} \in R'|}$$

Thus the system can determine the users' interest by matching with tags from each cluster. It is the proportion of times the specific user who used tag t_i in a cluster to annotate a resource $r_{m'} \in R'$ over the nubmer of times that all tags annotated by this user.

After ranking the users basing on calculated their interest to each tag cluster, we also need to match each resource to its closest clusters. Similar to calculate user's interest, we define it as:

$$rc(r_{m'}, c_k) = \frac{|\tau_{ijk} : t_i \in c_k, u_j \in U|}{|\tau_{ijk} : t_i \in T, u_j \in U|}$$

is the the proportion of times the specific resource which is annotated by tag t_i in a cluster over the number of times that all tags were used to annotate this resource.

These two proportions reflect the relationship between tag clusters and users or resources. It takes a value from zero to one. A value close to one represent a stronger relationship. While the proportion value of resource to clusters would be constant, the proportion value associate users and clusters would be different according to users' profile.

As the tag clusters paly a role of the links between resoueces and users who contributed them, the relationship of users and resources can be ranked. We define the relationship as:

$$rel(u_j, r_{m'}) = \sum_{c_k \in C} uc(u_j, c_k) * rc(r_{m'}, c_k)$$

is the sum of every product of two proportions over the total number of clusters. Thus we can link the users' interest to every resource in the subset.

In order to handling the query tag and return the resources reflect users' interest, we define a rank score as:

$$rs(q, u_j, r_{m'}) = sim(q, r_{m'}) * rel(u_j, r_{m'})$$

This is the final phase of the process of recommendation. As we combined the similarity between query tag and resources with the relationship of users and resources, we can compute each resource in subset and re-rank them

according to users' interest, finally return the top n resources to users as personalized recommendation.

5.2 Tag Recommendation

Another application of tag clusters is tag recommendation.

The system is able to recommend the most popular tags as a tag list to users. For each tag in tag space, we can count the number of times it has been annotated to the resources by tag frequency as we have denoted:

$$tf(t_i, r_m) := |\{(t_i, u_j, r_m) \in \tau : u_j \in U\}|$$

by which we can use the top n tags as popular tag recommendations.

Based on the predefined tags, users can choose some of the tags with high frequency of usage in the same or similar resources through a given interface, rather than using a individual tag.

For a given resource, we can calculate for all tags the number of times they have co-occured together with a certain resource. We define the score of co-occurrence as:

$$cotags = tf(t_i, r_m) \cdot tf(t_j, r_m)$$

then take the tags with the highest score of co-occurrence with the resource for the tag recommendation. With this application, the system can control the tag space and reduce the redundancy and ambiguity problem in annotation.

Conclusion

In this paper, we have mentioned the collaborative tagging systems, a collaborative method to manage and categorize resources based on tags or keywords. Comparing with traditional method where a hierarchical classification of resource is done by professional or author, the collaborative tagging systems is a web application that users contribute to add tags to share the web resources. This method has its special motivations and advantages.

For selfish or personal purposes, the tagging system allows users to develop a personal digital filing system that applies keywords and easily retrieves the annotated resources. For social communication, it allows users to express themselves and share resources with other system users. It reflects a common interest and provides a tool for searching informations collected by multiple users.

As a collaborative system allows users to freely add tags and participate in the process of content classification (without the effort involved by adding terms to a controlled vocabulary), one can add, change or remove a tag when its meaning varies over time. The system updates and reflects immediately the results of such changes without any need to wait for a long as required. The flat and distributed structure of collaborative tagging systems appears to be a main advantage. With this structure the systems can contain a rich relationships between resources, which would help the users' information retrieval.

At the same time, without a control of vocabularies results in many semantic

ambiguity and redundancy problems which is caused by users' freely tagging behaviors such as polysemy, synonymy, and different forms of spelling, etc. With these problems a seeker who are searching for informations may use different words and find out a limited searching result.

Thus we discuss several clustering techniques that can be used in collaborative tagging systems in order to overcome the redundancy and ambiguity problems and improve the searching result, such as: hierarchical agglomerative clustering, k-means clustering and spectral clustering based on graph theory.

No matter what clustering method we choose to use in tagging systems, the first thing we need to is calculating the similarity between tags. Both distance measure and similarity measure can be used for clustering algorithm. By using one of the two measures, we can clustering the tag set into partitions whose members have minimal distance value and the tags in different partitions have a maximal distance value. However, different distance or similarity measure would procure different clustering partitions that contain different tags.

Based on the similarity measure, K-means clustering is one of the simplest, efficient and automated clustering approach that we could utilise in tagging systems. But it takes a drawback that irrelevant tags would not be isolated from tag partitions.

The hierarchical agglomerative clustering is also a traditional clustering method that we can use in tagging system. It first aggregates the tags who have the maximal similarity value based on a threshold, and then turn to the less similar tags. With a certain level of similarity, we can get a better quality clustering partition without irrelevant tags.

The spectral clustering based on graph theory is now a popular method with high performance computing and dimensionality reduction in tagging systems. Tags have been denoted as vertices and the relationship between tags is represented as edges. By calculating the matrix of the similarity weight of

edges W and the degree of each vertex D , we can take the eigenvector corresponding to the second smallest eigenvalue of the graph Laplacian matrix $\mathcal{L} = D - W$ to cluster the tag set into two partitions.

Finally, with the results of clustering algorithm, tags are strong related and disambiguated within each cluster. The collaborative tagging systems is able to track more accurately the common interests of users by measuring users over tag clusters. Resources associated with tags clusters can also be captured for information researching. The system could provide a recommendation application to users for better usage.

The experiment of each clustering algorithm for tagging system would be done in further studies in order to improve the accuracy and efficiency of recommendation application.

Bibliografia

- [1] Louise F. Spiteri, Structure and form of folksonomy tags: The road to the public library catalogue, 2007, Webology, Vol. 4, Nr. 2, <http://www.webology.org/2007/v4n2/a41.html>
- [2] Adam Mathes, Folksonomies - Cooperative Classification and Communication Through Shared Metadata, 2004, <http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html>
- [3] Golder and Huberman, The Structure of Collaborative Tagging Systems, 2006, <http://arxiv.org/ftp/cs/papers/0508/0508082.pdf>
- [4] Emilee Rader, Rick Wash, Influences on Tag Choices in del.icio.us, 2008, <http://bierdoctor.com/papers/delicious-csw-logistic+simulations.pdf>
- [5] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, and Robin Burke, Personalization in Folksonomies Based on Tag Clustering, 2008, Proceedings of the 6th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems, <http://www.aaai.org/Papers/Workshops/2008/WS-08-06/WS08-06-005.pdf>

- [6] Grigory Begelman, Philipp Keller, Frank Smadja, Automated Tag Clustering: Improving search and exploration in the tag space, 2006.
- [7] Rick Wash, Emilee Rader, Public Bookmarks and Private Benefits: An Analysis of Incentives in Social Computing, 2007, Proceedings of the ASIS&T Annual Meeting.
- [8] Clay Shirky, Ontology is Overrated: Categories, Links, and Tags, 2005, http://www.shirky.com/writings/ontology_overrated.html
- [9] Wolfgang G. Stock, Folksonomies and science communication, 2007, Journal: Information Services & Use, Volume 27 Issue 3.
- [10] Alireza Noruzi, Folksonomies: Why do we need controlled vocabulary, 2007, Webology, Volume 4, Number 2, <http://www.webology.org/2007/v4n2/editorial12.html>
- [11] www.wikipedia.org
- [12] Edith Speller, Collaborative tagging, folksonomies, distributed classification or ethnoclassification: a literature review, 2007.
- [13] Paul Anderson, What is Web 2.0? Ideas, technologies and implications for education, 2007, JISC Technology and Standards Watch.
- [14] Josef Kolbitsch, Aspects of Digital Libraries, 2007, http://www.iicm.tu-graz.ac.at/iicm_thesis/jkolbitsch1.pdf
- [15] Hesham Allam, Social Tagging as a Knowledge Organization and Resource Discovery Tool, 2010, Dalhousie Journal of Interdisciplinary Management, Vol 6.
- [16] http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/

- [17] Pankaj Jajoo, Document Clustering, 2008.
- [18] YANG SONG, Automatic Tag Recommendation Algorithms for Social Recommender Systems, ACM Transactions on the Web, 2009, <http://research.microsoft.com/pubs/79896/tagging.pdf>
- [19] Valentin Robu, Harry Halpin, Hana Shepherd, Emergence of Consensus and Shared Vocabularies in Collaborative Tagging Systems, 2009, Journal ACM Transactions on the Web (TWEB) TWEB Volume 3 Issue 4.
- [20] Ulrike von Luxburg, A Tutorial on Spectral Clustering, 2007, Statistics and Computing, Vol. 17, No. 4.
- [21] Scott Whitey, Padhraic Smythy, A Spectral Clustering Approach To Finding Communities in Graphs, 2005.
- [22] Satu Elisa Schaeffer, Graph clustering, 2007, computer science review (27-64), <http://dollar.biz.uiowa.edu/~street/graphClustering.pdf>
- [23] Jianbo Shi, Jitendra Malik, Normalized Cuts and Image Segmentation, 2000, IEEE Transactions on pattern analysis and machine intelligence, Vol.22, No.8, <http://www.cs.berkeley.edu/~malik/papers/SM-ncut.pdf>
- [24] Giovanni Rossi, Pseudo-Boolean clustering, 2010, Technical Report UBLCS
- [25] M. E. J. Newman, M. Girvan, Finding and evaluating community structure in networks, 2004, Physical Review E, Vol. 69, No. 2, http://arxiv.org/PS_cache/cond-mat/pdf/0308/0308217v1.pdf

- [26] Inderjit S. Dhillon, Co-clustering documents and words using Bipartite Spectral Graph Partitioning, 2001, In Knowledge Discovery and Data Mining.
- [27] Eirini Giannakidou, Vassiliki Koutsonikola, Athena Vakali, Ioannis Kompatsiaris, Co-Clustering Tags and Social Data Sources, 2008, The Ninth International Conference on Web-Age Information Management, <http://mklab.itι.gr/files/GKVKdraft.pdf>
- [28] Pankaj K. Agarwal, Clustering & classification, 2003, CPS260/BGT204.1 Algorithms in Computational Biology.
- [29] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, Robin Burke, Personalized Recommendation in Social Tagging Systems Using Hierarchical Clustering, 2008, In Proceedings of the 2008 ACM conference on Recommender systems (2008), pp. 259-266.
- [30] Jun Tang, Improved K-means Clustering Algorithm Based on User Tag, 2010, JCIT 5(10): 124-130, http://www.aicit.org/jcit/ppl/016_JCIT3-699042.pdf
- [31] Huazhong Ning, Wei Xu, Yun Chi, Yihong Gong, Thomas Huang, Incremental Spectral Clustering With Application to Monitoring of Evolving Blog Communities, 2007.
- [32] Ingyu Lee, Byung-Won On, An effective web document clustering algorithm based on bisection and merge, 2011, Artificial Intelligence Review, Volume 36, Number 1.
- [33] Newman, Fast algorithm for detecting community structure in networks, 2003, Phys-RevE.69.066133, http://arxiv.org/PS_cache/cond-mat/pdf/0309/0309508v1.pdf

- [34] Andriy Shepitsen, Noriko Tomuro, Search in Social Tagging Systems Using Ontological User Profiles, 2009, International Conference on Weblogs and Social Media - ICWSM , 2009, <http://facweb.cs.depaul.edu/noriko/papers/ICWSM09-shepitsentomuro.pdf>
- [35] Lixin Han, Guihai Chen, HFCT: A Hybrid Fuzzy Clustering Method for Collaborative Tagging, 2007, In Proc. International Conference on Convergence Information Technology.
- [36] Lixin Han, Hong Yan, A fuzzy biclustering algorithm for social annotations, 2009, Journal of Information Science, vol. 35 no. 4, 426-438
- [37] Yihong Dong, Yueting Zhuang, Fuzzy hierarchical clustering algorithm facing large databases, 2004, Intelligent Control and Automation
- [38] Yihong Dong, Yueting Zhuang, Ken Chen, Xiaoying Tai, A hierarchical clustering algorithm based on fuzzy graph connectedness, 2006, Fuzzy Sets and Systems Volume 157, Issue 13, Pages 1760-1774
- [39] M. Ames and M. Naaman. Why we tag: Motivations for annotation in mobile and online media. In CHI '07, 971–980, 2007.