WEB IMAGE CO-CLUSTERING BASED ON TAG AND IMAGE CONTENT FUSION

Jie Chen [1, 2, 3], Jianlong Tan [2, 3], Xiangzhou Yin [2, 3] , Hao Liao [2, 3]

1. Beijing University of Posts and Telecommunication, Beijing, China
2. Institute of Computing Technology Chinese Academy of Sciences, Beijing, China
3. National Engineering Laboratory for Information Security Technologies, Beijing, China
chenjie@software.ict.ac.cn

Abstract

In Web 2.0 applications, users always label digital images using textual descriptions, which are also called tags. As a result, a web image usually carries both tag and visual content information. In order to improve the retrieval performance of web images, in this paper, we propose an error-driven fusion co-clustering algorithm, which combines images' tags, visual contents together for analysis. Experimental results demonstrate that our algorithm outperforms other simple clustering methods.

Keywords: error-driven fusion; co-clustering; image retrieval;

1 Introduction

Under the Web 2.0 scenario[1], web data has many distinctive features from data in conventional databases. Web data usually exhibits the following characteristics [2]: huge in amount, distributed, heterogeneous, and dynamic. So, it becomes very difficult for user to retrieve their interested information efficiently under the Web 2.0 context. This becomes even harder when it comes to the web image retrieval tasks due to the semantic gap problem. There has been a line of work recently proposed using the clustering methods to analyze and organize images, most of which are based on either the image features, or other auxiliary information, such as the annotations [3, 4].

Existing methods primarily perform image clustering by using tag information or the visual features extracted from the web images. However, under Web 2.0 context, users usually label digital image resources using arbitrary tags. This makes many tags of the web images unable to precisely describe their semantic information. Moreover, most of the state-of-the-art visual features extracted from the web images are also incapable to tackle the semantic gap problem. Indeed, in Web 2.0 applications, we need to co-cluster different objects simultaneously. Therefore, many co-clustering methods were proposed and widely used in grouping together objects from different datasets [5, 6]. Following this logic, in this paper, we propose an error-driven fusion co-clustering approach to combine both the web images' tag information and visual features to boost the performance of the web image retrieval tasks. More specifically, we propose a new image co-clustering method by first clustering images w.r.t. their tags, and then clustering the outliers (which are identified in the previous co-clustering step) w.r.t. their visual contents. In our case, co-clustering is able to cluster similar image tags and visual contents together simultaneously.

The design of our approach is shown in Figure 1. Our approach consists of four steps. In the first step, we use a co-clustering method to cluster images w.r.t. their tag information. In the second step, for each cluster, we assign a threshold value to split all the images into the "correct" classified set and the "outlier" set. In the third step, to each image in the outlier set, we use the co-clustering method once more w.r.t. both the image's content and tag information. At last, all the clustering results acquired in the third step are ensembled together using the average weighting mechanism, to adjust the tag co-clustering results in the first step. By doing so, the co-clustering results are able to incorporate more information (i.e., both the tag and the visual content information) to boost the retrieval performance.

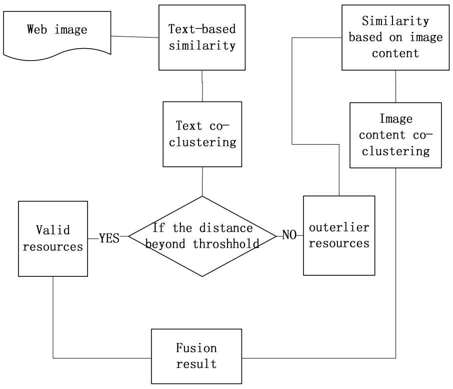

Figure 1 The overview of our approach

The rest of this paper is organized as follows: section 2 defines the basic problem formulation. Section 3 analyzes the employed similarity measures and proposes an error-driven fusion co-clustering method. Section 4 provides experiments and we conclude this paper in Section 5.

2 Problem formulation

We give some notations first. Consider a web image which is described by many arbitrary tags and some visual content information. Table 1 summarizes all the notations used in this paper.

Table 1 Basic Symbols Notation

Symbol	Definition
m, n ,l ,d, n'	Number of picture keys, resources, tags, attributes and outlier resources(respectively)
R,T	Resources' set $R=\{r_1, r_2, \cdots, r_n\}$, tags' set $T=\{t_1, t_2, \cdots, t_l\}$
AS, PKEY	Attributes' set $AS=\{at_1, at_2, \cdots, at_d\}$, Image Keys' set $PKEY=\{pf_1, pf_2, \cdots, pf_m\}$
P	Images' set $P=\{p1, p2, \cdots, p_{n'}\}$
$f(r_i, t_j ,)$	Annotation function of tag t_j to resource r_i

Function *f* is used to determine whether a tag t_j, $(j=1 \cdots l)$ has been used for the annotation of resources r_i, $i=1 \cdots n$, if t_j is an annotation tag for r_i, we let $f(r_i, t_j)=1$, otherwise, 0.

Definition 1 (RESOURE'S REPRESENTATION) Each resource $r_i \in R$, $i=1 \cdots n$, is represented by aggregating the tags assigned to it and it can be identified as follows:

$$r_i = \{\bigcup t_x\},\ \forall t_x \in T : f(r_i, t_x) = 1$$

In practice, the number of tags or images represents a specific resource may grow in large scale and thus we need to employ a selection process of the most distinguishing tags or images which will form the resources' attribute set *AS* and picture key set *PKEY*. In our approach we use the *d* most frequent tags to form the *AS* set and *m* most typical images to form the *PKEY,* which will guide our clustering process.

Definition 2 (THE ATTRIBUTE SET) Given the $T=\{t_1, \cdots, t_l\}$ set of tags, we define the attribute set $AS=\{at_1, \cdots, at_d\}$: $AS \in T$, and *AS* contains the *d* most frequent tags $t_x \in T$.

Definition 3 (THE IMAGE KEY) Given the $P=\{p_1, p_2, \cdots, p_{n'}\}$ set of images, we define the image key set. $PKEY=\{pf_1, pf_2, \cdots, pf_m\}$, $PKEY \in P$, and *PKEY* contains the *m* most typical images.

Problem 1 (RESOURE-ATTRIBUTES-IMAGE KEYS CO-CLUSTERING) Given a set *R* of *n* resources, a set of *AS* of *d* attributes, a set of *PKEY* of *m* image keys, an integer *k* and similarity function, find a set *C* of *k* subsets of resources, attributes and image keys, $C=\{C_1, \cdots, C_k\}$, C_i $(1 \leqslant i \leqslant k)$ such that

$$\sum_{x=1}^{k}\Big(\sum_{r_i, at_j \in c_x} Similarity(r_i, at_j) + \sum_{p_i, pf_t \in c_x} Similarity(p_i, pf_t)\Big),$$

$i =1, \cdots, n'$, $j=1, \cdots, d$, $t=1, \cdots, m$, is maximized.

The *Similarity* function must be defined in a way to sufficiently capture the distance between each resource and each attribute, as well as the distance between each image which represents resource and image keys.

3 An error-driven algorithm based on tag and picture fusion

3.1 similarity measure

As discussed above, each resource can be represented by a set of tags (Definition 1). Thus, finding the relationships between a resource and an attribute indicates measuring the similarity between tags and attributes. Apply social and semantic similarity jointly to describe the similarity between tags and attribute [6]. So the social similarity which is based on tagging information and semantic similarity which based on mapping technonique between two tags t_x and t_y. We can define as follows:

$$SoS(t_x, t_y) = \frac{\sum_{i=1}^{n} r_i : (r_i, t_x) \in A \vee (r_i, t_y) \in A}{\max\left(\sum_{i=1}^{n} r_i : (r_i, t_x) \in A, \sum_{i=1}^{n} r_i : (r_i, t_y) \in A\right)} \quad (1)$$

$1 \leqslant x, y \leqslant l$, $r_i \in R$.

$$SeS(t_x, t_y) = \frac{2 * depth(PTA)}{depth(t_x TP) + depth(t_y TP) + 2 * depth(PTA)} \quad (2)$$

We believe that two concepts are similar if they possess the same root in *Wordnet* [8]. $depth(t_x TP)$ and $depth(t_y TP)$ is the number of nodes on the path from t_x or t_y to the common node, $depth(PTA)$ is the number of nodes on the path from the common node to root.

Thus, we define the *Similarity Score SS* between two tags t_x and t_y in terms of both their social and semantic similarity as in Eq. (3) :

$$SS(t_x, t_y) = w * SeS(t_x, t_y) + (1 - w) * SoS(t_x, t_y) \quad (3)$$

where *w* is a weight factor, and $w \in [0,1]$. Given the *Similarity Score,* we define the *similarity* function between resources and attributes. The *similarity* function is defined as the maximum Similarity Score between every tag assigned to the resource and the attributes. Thus:

$$Sim(r_i, at_j) = \max_{x=1\cdots|r_i|}\{SS(t_x, at_j)\}, \mathrm{r_i} \in \mathrm{R}, \mathrm{t_x} \in \mathrm{r_i}, at_j \in AS$$

The value of similarity function between each of the *n* resource and *d* attributes are then used to formulate an *n*×*d* table *RA* as follow:

$$RA(i, j) = Sim(r_i, at_j);\ i=1,\cdots,n;\ j=1,\cdots,d \quad (4)$$

A resource is also represented by an image visual content. Thus, finding the relation between a resource and an image key also indicate capturing the similarity between the resources' image content and the image key. For example, we only need to calculate the distance between them for capturing similarity. In our paper, color histogram is used to execute distance calculation between images *X* and *Y* according to Correlation coefficient, CHISQR, INTERSECT and Bhattacharyya.

(1) Correlation coefficient:

$$\rho = \frac{\sum XY - \frac{\sum X \sum Y}{n}}{\sqrt{[\sum X^2 - \frac{(\sum X)^2}{n}][\sum Y^2 - \frac{(\sum Y)^2}{n}]}}; -1 \le \rho \le 1$$

(2) CHISQR: $\chi^2 = \sum \frac{(X-Y)^2}{(X+Y)}$

(3) INTERSECT：$X \cap Y = \sum \min(X, Y)$

(4) Bhattacharyya:

$$Bhattacharyya = \sqrt{1 - \frac{\sum \sqrt{XY}}{\sqrt{\sum X \sum Y}}}$$

(5)

Besides, in order to effectively compare the experiment results, we also use SIFT [9] to capture similarity between images. Similarity between Pictures *A* and *B* is formulate: *sim*(*A*,*B*)=*N/N*$_a$, where *N* is the number of match keypoints and N_a represents the number of keypoints in *A*. Considering asymmetric similarity between two images, we can describe similarity between B and A using *sim*=(*B*, *A*)=*N/N*$_b$, where N_b is the number of keypoints in *B*. So the similarity between each of the *n'* outlier resources and *m* picture keys are used to formulate the *n'*×*m* table *PF* as follow:

$$PF\ (i, j) = Sim(p_i, pf_j);\ i=1\cdots n',\ j=1\cdots m \quad (5)$$

3.2 Dataset bipartite graph representation

Since our method handles the simultaneous clustering of resources, attributes and image keys, we need to use a data structure that is able to efficiently represent their relations. The Bipartite graph can represent the relations and has already been used for describing co-clustering problem [10]. Let us consider the bipartite graph *G*=(*R, AS, PKEY, E*) present in Figure 2, where *R*={r_1, r_2, r_3} the set of sources, *AS*={at_1, at_2} the set of resources, *PKEY*={pf_1, pf_2} the set of image keys and *E*={{r_i, at_j }: $r_i \in R$, $at_j \in AS$ or {r_i, pf_j}: $r_i \in R$, $pf_j \in PKEY$}. In the case of the bipartite graph of Figure 2, its two-partitioning depicted in Figure 3 would result in the maximization of the sum of similarities between the elements belonging to the same clusters, while the sum of similarities between the elements of different clusters would be minimized. In other words, we are looking for a *k*-partitioning (*C1, C2,…,Ck*) of the graph, such that

$$\sum_{r_i \in C_x}\Big(\sum_{at_j \in C_y} Similarity(r_i, at_j) + \sum_{pf_l \in C_y} Similarity(r_i, pf_l)\Big) \quad (6)$$

x,*y*=1,⋯,*k* and *x* ≠ *y* can be minimized.

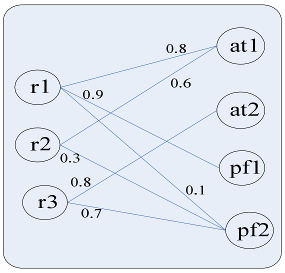

Figure 2 Data representation

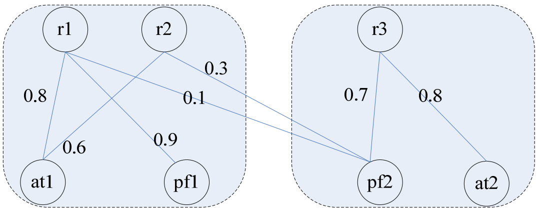

Figure 3. Cut of the Bipartite Graph

3.3 The error-driven fusion co-clustering

The error-driven fusion co-clustering problem can be transformed to the conventional bipartite graph partition problem, which can be solved by the spectral graph clustering method [11]. Spectral graph clustering algorithms rely on the eigenstructure of a similarity matrix to partition points into disjoint clusters. More specifically, eigenvector decomposition is performed on the similarity matrix traditional clustering techniques, such as *k*-means, may be an applied to the defined by the eigenvectors. As proven in [11], we can execute co-clustering by normalizing table $NRA=D_r^{-1/2} RA\ Dat^{-1/2}$ and eigenvector decomposition:

$$SV = \begin{bmatrix} D_r^{-1/2} L_r \\ D_{at}^{-1/2} R_{at} \end{bmatrix}$$

The D_r and D_{at} are the diagonal degree tables of sources and attributes, respectively. L_r denote the $n \times k$ table of the left singular vectors and R_{at} the $d \times k$ table of the right singular vectors of *NRA* table. Then running a typical clustering algorithm on *SV* will result in *k* clusters. We use the same way to solve co-clustering based on image content about similarity function *PF* (Equation 5).

Now we introduce the error-driven procedure. We use tags to represent resources, and then the co-clustering method is executed on tag similarity function. So some clusters are obtained with containing both resources and attributes. A satisfactory threshold is used to decide the distance from all the data to the clustering center. We split each cluster into two different types: the valid points and outlier points, according to the given threshold. In a tag cluster, if the distance between a data point to the clustering center is lower than the threshold, then we take it as an outlier point, otherwise we take it as a valid point. At this time, we can use image content to represent all the outlier set, and then image-based co-clustering is applied to cluster outlier resources. In other words, we cluster outlier set once more, some image clusters appear in such a way. Finding out the closest point to center in the image clusters, if the closest point belongs to i^{th} (i=1,⋯,k) tag cluster, we can take all points in the image clusters into i^{th} valid points. So the tag cluster is refined by using image co-clustering on outlier resources. The fusion cluster is better than tag cluster in terms of performance. In Figure 4, we demonstrate the entire processes for Erro-driven fusion for a specific example:eight points (ui,i=1⋯8) for two-group partitioning.

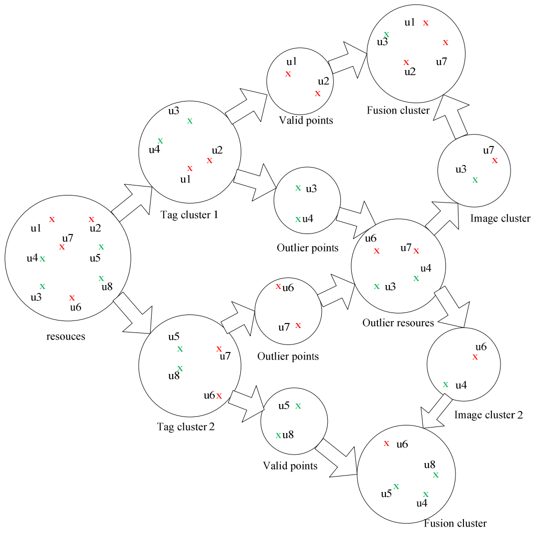

Figure 4 Error-driven fusion process

The steps of the algorithm are as follows:

Algorithm 1 The Error-driven Fusion Co-clustering Algorithm

Input: The set of *n* web images, a integer *k*, a real number w where $w \in [0, 1]$

Output: A set $C=\{C_1,\ldots,C_k\}$ of *k* subsets consisting of tags and image content information.

1. Calculate similarity score based on tags: $SS= w*SoS+(1-w)*SeS$
2. Obtain the similarity function: *RA*= Similarity(*SS*)
3. Calculate the degree tables D_r and D_{at}, then we form the table $NRA= Dr^{-1/2}RA\ Dat^{-1/2}$
4. Apply singular value decomposition on *NRA* to obtain L_r and R_{at}.
5. Integrate D_r, D_{at}, Lr, Rat in *SV*.
6. Run k-means on *SV* and obtain *k* tag clusters.
7. Select threshold for each tag cluster according to distance and form outlier recourses
8. Calculate similarity between outlier resourses based on image visual content and obtain similarity function *PF*.
9. Apply the same method on *PF* according to step 3, 4, 5, 6 and we can obtain *k* image clusters.
10. Find out the closest point to center in the image clusters, if the closest point belongs to i^{th} (i=1, ⋯ ,k) tag cluster, we can take all points in the image clusters into i^{th} valid points. We are able to obtain fusion clusters including both tags and image contents.

4 Experiments and results

4.1 Experimental setup

In order to evaluate our algorithm, we use a dataset from the Flickr[1] data set(Flickr photo-sharing system: http://www.flickr.com). We use seven scenes, each of which consists of 1000 images regarding the cityscape, seaside, mountain, roadside, landscape, sport, and locations (totally 7000 images are used in our experiments). These images are composed of both the visual content and tag information. Besides, we restrict the AS's size to d = 70 tags and PF = 35. We use precision and recall to measure clustering results.

4.2 Image co-clustering results evaluation

Experiments are under the given values k=7, and w=0.5 (as described in [6] and we have proved that this value is the optimal one). As described in subsection 3.3, some thresholds are used for each cluster after tag-based co-clustering. We execute the tag-based co-clustering experiments and obtain optimal thresholds by experience. The precision and recall results for each of the obtained clusters are depicted in Tables 2 and 3. Two lines in Tables

2 and 3 list the results of the error-driven fusion and the flat tag co-clustering (*w*=0.5) method respectively. The results show that in most cases, the extracted clusters are good.

In the following part, we measure the precision and recall using the F-Measure metric which is a widely used to evaluate clustering performance. The F-measure value fluctuates in the interval [0…1] with higher values indicating a better clustering. Figure 5 lists the F-measure value for all the obtained clusters. When *k*=7, by the error-driven fusion co-clustering and flat tag-based co-clustering. As we can observe, the F-Measure for our method is mostly higher than flat tag-based co-clustering, because we also consider the image content when co-clustering web image based on tags. Thus, the current error-driven fusion method is more efficient than the traditional flat tag co-clustering methods. In fact, compared to the work reported in the reference [6], our results achieve better results. In summary, our method is able to provide efficient fusion co-clustering results.

Table 2 Clusters' threshold

Cluster (k=7)							
Cluster number	1	2	3	4	5	6	7
Threshold	0.013	0.012	0.01	0.01	0.014	0.01	0.007

Table 3 Clusters' Precision

Cluster (k = 7)							
	1	2	3	4	5	6	7
Error-driven	0.61	0.71	0.96	0.97	0.97	0.92	0.72
w=0.5	0.51	0.70	1	0.96	0.97	0.91	0.75

Table 4 Clusters' Recall

Cluster (k = 7)							
	1	2	3	4	5	6	7
Error-driven	0.86	0.67	0.61	0.9	0.96	0.95	0.53
w=0.5	0.98	0.65	0.59	0.96	0.97	0.97	0.48

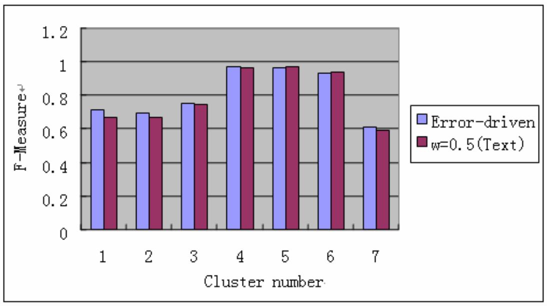

Figure 5 Clusters' F-measure

5 Conclusions

In this paper, we propose an error-driven fusion method for effectively co-clustering a large number of web images. The co-clustering method combines both the images' contents as well as the images' tags to yield good clusters. Experimental results demonstrate that the proposed method performs better than many traditional flat co-clustering methods which merely rely on either the images' tag information, or content information. Moreover, our method is able to handle the tag invalidity problem of web images.

Acknowledgements

This work was funded by The National Basic Research Program (973 programs)
(2007CB311100)

References

[1] T. O'Reilly. What is web 2.0, September 2005.http://www.oreillynet.com/pub/a/oreilly /tim/news/2005/09/30/ what-is-web-20. html

[2] Zhang, Y., J.X. Yu, and J. Hou, Web Communities: Analysis and Construction. 2006, Berlin Heidelberg: Springer.

[3] X. He, D. Cai, H. Liu, and J. Han. Image clustering with tensor representation. In proc. of ACM MM, 2005, 132–140.

[4] H. Wang, S. Yan, T. Huang, and X. Tang. Maximum unfolded embedding: formulation, solution, and application for image clustering. In proc. of ACM MM, 2006, 45–48.

[5] I.Dhillon, S. Mallela, and D. Modha. Information-theoretic co-clustering. Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, (KDD'03), 2003, 89-98.

[6] Eirini Giannakidou, Vassiliki Koutsonikola, Athena Vakali, et al. Co-Clustering Tags and Social Data Sources. The Ninth International Conference on Web-Age Information Management. 2008, 317-324.

[7] C.Fellbaum. WordNet, an electronic lexical database. The MIT Press,1990.

[8] David G.Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision: VOL 60, NO 2, 2004.

[9] I.S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. Proc. of the 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data mining, 2001,269-274.

[10] S. Loong and S. Mishra. Spectral graph partitioning analysis of in vitro synthesized RNA structural folding. Proc. of 2006 Int. Workshop on Pattern Recognition in Bioinformatics,2006,81-9.