2015 IEEE International Conference on Multimedia Big Data

Integrating Visual and Textual Features for Web Image Clustering

D. S. Xia, Z. Q. Xiang, Y. X. Zou*
ADSPLAB/ELIP, School of Electronic and Computer Engineering
Peking University
Shenzhen 518055, China
*zouyx@pkusz.edu.cn

***Abstract*—With the explosive growth of Web and tremendous development of digital image processing technologies, the applications of Web image have attracted much attention, such as the Web image retrieval. Since the Web images are often with some related text tags, making use of both visual and textual features of Web image will help improving the accuracy of the Web image clustering. Researches show that Web image clustering methods, such as graph partitioning models and hypergraph partitioning models, didn't make use the relations between texts and image simultaneously. In this paper, we explore to take both visual and textual features into account for Web image clustering by building a graph model and develop a novel iterative clustering method. With *K* clusters initialized, we calculate the occurrence frequency of each visual/textual feature over the *j*-th cluster (*j* = 1, 2, …, *K*), which is used to measure the significance of the feature for the *j*-th cluster. Then the likelihood of each image, which belongs to the *j*-th cluster, can be determined accordingly. Furthermore, a mixture model is built for the predicted feature linked to each image and the EM algorithm is adopted to get *K* component parameters which describe posterior probabilities of all clusters for each image. Then two *K*-dimensional vectors consisting of component parameters will be used to describe the image and adjust the cluster index of it. Several experiments have been performed with MIR-Flickr25K and IAPR TC-12 Benchmark datasets and the performance of the proposed Web image clustering algorithm is superior to that of the compared algorithm.**

Keywords-Web image clustering; visual and textual features; ranking functions; mixtual model

I. INTRODUCTION

Nowadays, the rapid development of Web and multimedia processing technologies has brought a great promotion on the increase of Web images and some other media [1]. As a critical technology of the analysis and mining on the multimedia, clustering of Web images has been used in many applications of Web images. For example, the performance of Web image retrieval can be improved by the clustering algorithms. At the current time, most famous Web images search engines, such as Google Images[1] and Baidu Image[2], are performed based on matching the textual keywords, viz. tags. However, the multiple meanings of the keywords have always brought negative effect on the image search results. For example, the word "apple" could mean the fruit "apple" or the company "Apple". So when we search for this word, the search results would be returned in a mixture of different topics, as shown in Fig. 1. And things may get worse when one of the topics is quite overwhelming and happens to be not the user desiring one. If we query with more specific words, it may introduce some more noisy topics. One efficient solution to this problem is to cluster the Web images in the search results.

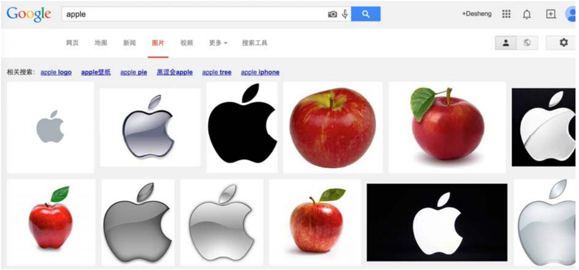


Figure 1. The multi-topics of Google images retrieval results by searching keyword "apple" on Dec. 12, 2014

Traditionally, the clustering of Web images adopts the clustering algorithms based on calculating the similarities between the low-level visual feature vectors of images [2, 3]. Here, the low-level visual feature vectors may be color features, Sift features, etc. The performance of those methods is usually limited by the drawbacks of extracted features. To improve the performance of clustering, researchers introduce textual features in their algorithms [4, 5]. Generally, the textual features are words or textual tags extracted from the surrounding texts of the Web images.

Motivated by the ranking based clustering algorithms for the multi-type relational data [6], we propose a new iterative image clustering algorithm for Web image clustering. In our algorithm, before the clustering process, all the Web images should be initialized into *K* clusters randomly. Then, we calculate the ranking scores on each subgraph induced by the clusters and either type of visual and textual features, which could be viewed as probability distributions of the features on the each cluster. Next, based on the ranking scores of visual and textual features respectively, we build two mixture models for each image, which describe the probabilities of relations between image and visual or textual features. After that, a 2*K* dimensional vector consisting of the component coefficients can be calculated for adjusting the current clusters. All the above steps are repeated until the clusters are converged. The main contributions of this paper are as follows:

1. Unlike the graph partitioning strategies, we apply the ranking based clustering framework into the Web image clustering problem. The ranking based clustering algorithm calculates the ranking distributions which could give better understanding of the relations between images and their features.

[1] https://images.google.com
[2] http://image.baidu.com

978-1-4799-8688-0/15 $31.00 © 2015 IEEE
DOI 10.1109/BigMM.2015.35

2. We propose a new probability distribution calculating method based on ranking functions to take benefits of the complicated relations among the features. Our method builds more direct relations between the Web images and their features hence the performance of the Web image clustering is improved.

This paper is organized as follows. In Section II, a brief review of the related research works will be listed. Then in Section III, we give some precise definitions and descriptions for the problem and models. In Section IV, the details of the algorithm are described. And in Section V, some experiments have been carried out. At last, we conclude our study in Section VI.

II. Related Works

Clustering Web Image has been researched for a long time. In the early days, much work has been focus on clustering by their low-level visual features. Here, the low-level visual features denote the features extracted directly from the image data, such as Sift features, Giber wavelet features and so on. For example, Gordon [2] and Yang [3] extracted feature vectors to represent the images and then perform the clustering algorithms on the feature representations. However, due to the "semantic gap" between the low-level visual features of images and high-level perception of users, those methods cannot obtain the expected cluster results.

Since the web images are often surrounded by some semantic related texts, researchers began to extract the textual features from the surrounding texts. The textual features are usually the critical words or text tags. There has been some excellent work proposed in the last few years [7-9]. In particular, D. Cai [4] proposed a graph model to co-cluster the web images and their represented features. Their algorithm separates the textual features and visual features and was executed in a two-step process. The images were firstly clustered into different semantic groups by employing the textual and link features. This step was then followed by visual feature-based clustering of images in each semantic group. Then considering the errors in the results of the first step may be magnified in the following process, Gao et al [5] proposed another clustering framework to simultaneously integrate both the visual and textual features for clustering. In their work, spectral clustering was applied and iteratively using semi-definite programming to cluster. Similarly, Rege [10] proposed another clustering algorithm Consistent Isoperimetric High-Order Co-clustering (CIHC) with using isoperimetric theorem to integrate both visual and textual features at same time.

Since the graph models are widely used in the Web image clustering, hypergraph models are also introduced to cluster the Web images. For example, Zhou [11] and his team proposed a hypergraph partitioning model. Based on the hypergraph models, Wu [12] proposed a clustering method based on the random walk and that obtained great performance especially on large dataset.

III. Model Formulations

In this section, we propose an undirected graph model for integrating the visual and textual features and describing the relations among them (Fig. 2). In the graph $\boldsymbol{G} = \{\boldsymbol{V}, \boldsymbol{E}\}$, $\boldsymbol{V}$ denotes the set of vertexes in the graph and $\boldsymbol{E}$ denotes the set of edges. The vertex set $\boldsymbol{V}$ can be divided into three disjoint subsets $\boldsymbol{I}$, $\boldsymbol{T}$, $\boldsymbol{F}$, which satisfy $\boldsymbol{V} = \boldsymbol{I} \cup \boldsymbol{T} \cup \boldsymbol{F}$. The subset $\boldsymbol{I} = \{i_1, i_2, ..., i_m\}$ denotes the whole image set which has m images in total and the i_j (j=1,2,...,m) denotes the j-th image in the image set; the subset $\boldsymbol{T} = \{t_1, t_2, ..., t_n\}$ contains all n textual features which are usually the words or tags extracted from the surrounding texts of the images; and the subset $\boldsymbol{F} = \{f_1, f_2, ..., f_p\}$ contains the p visual features of images.

As we can see from the definition and reality of Web images, we can assume that there are no direct relations between visual and textual features. So the edge set $\boldsymbol{E}$ can also be divided into three subsets. The subset $\boldsymbol{E}_1$ denotes relations between images and their visual features. The subset $\boldsymbol{E}_2$ denotes the relations between images and their textual features. And the last subset $\boldsymbol{E}_3$ denotes the intra-relations among the different tags.

Let $\boldsymbol{A}$ be an m by p matrix to denote the weighted relations between $\boldsymbol{I}$ and $\boldsymbol{F}$. The entry $\boldsymbol{A}(l, j)$, where l = 1, 2, ..., m and j = 1, 2, ..., p, denotes the value of the j-th visual feature of the l-th image. Let $\boldsymbol{B}$ be an m by n matrix to denote the weighted relations between $\boldsymbol{I}$ and $\boldsymbol{T}$. The entry $\boldsymbol{B}(l, j)$, where l = 1, 2, ..., m and j = 1, 2, ..., n, denotes the weight of the j-th textual feature of the l-th image. The weights usually can be denoted by the frequency of the terms in the surrounding text of images. Let $\boldsymbol{C}$ be an n by n symmetric matrix to denote the weighted intra-relations among the textual features in $\boldsymbol{T}$. Here, the intra-relations can be described by the co-occurrence between different words or tags.

Then, we can get the whole weighted adjacency matrix $\boldsymbol{J}$ of the graph as follows.

$$\boldsymbol{J} = \begin{bmatrix} 0 & \boldsymbol{A}^T & 0 \\ \boldsymbol{A} & 0 & \boldsymbol{B} \\ 0 & \boldsymbol{B}^T & \boldsymbol{C} \end{bmatrix} \tag{1}$$

Particularly, during the text analysis, not only the inter-relations between textual features and images could be calculated, but also the weights of intra-relations among the

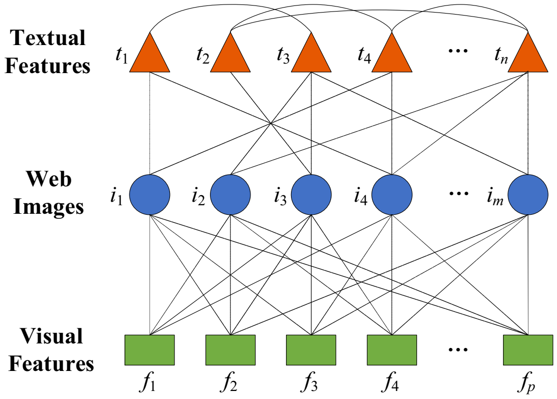


Figure 2. The graph model for the images and their textual features and visual features

textual features could be obtained. For most graph models, they model the images and the features as a bipartite graph or a tripartite graph, where those intra-relations have not been taken into account. So here we take the n by n matrix $\boldsymbol{C}$ to describe the relations among the textual features as described above.

Our problem is trying to cluster all the elements i_j (j=1, 2, ..., m) in the image set $\boldsymbol{I}$ into $\boldsymbol{K}$ clusters $\boldsymbol{C}_k$ (k=1, 2, ..., K).

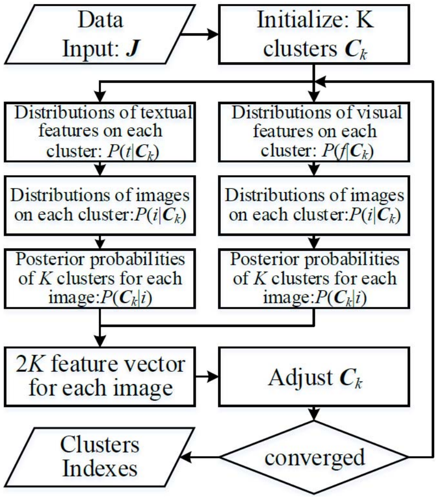
In addition, let the vector $\boldsymbol{r}$ be the ranking scores. If the ranking scores are calculated on the subgraph, the ranking scores can be regarded as the conditional ranking scores. For example, the ranking scores calculated on the subgraph induced by cluster $\boldsymbol{C}_k$ are regarded as conditional ranking scores on $\boldsymbol{C}_k$ and written like $\boldsymbol{r}_{x|\boldsymbol{C}_k}$ where x denotes any feature in $\boldsymbol{T}$ and $\boldsymbol{F}$.

IV. Iteration Clustering Algorithm

In this Section, the details of the proposed iteration clustering algorithm are presented.

Intuitively, data clustering aims to discover the similarity or homogeny among the whole dataset. The Web images in the same cluster would be similar in their visual features or share some identical text tags. Otherwise, the Web images in the different cluster may differ very much in their visual features or hardly share any same text tags. In other words, the different distributions of features on each cluster $\boldsymbol{C}_k$ could be viewed as a measurement for the cluster $\boldsymbol{C}_k$. In fact, for each image, the distribution of features could be viewed as a mixture model over K distributions of the features on each cluster. So, according to the idea above, the images may be described by the parameters of mixture models, which means that we could turn the visual and textual features into a new feature space. So, we try to calculate the probabilities of every textual feature and every visual feature for each cluster to build the mixture models. Then, new features would be extracted from the mixture models for each image. And at last, the clusters would be adjusted by the new features to get a better clustering result.

As visual features and textual features are not homogenous or related directly, it is reasonable to assume that visual features and textual features are independent. Therefore, the distributions may be calculated separately. Fig. 3 shows the flowchart of the proposed algorithm.



Data Input: $\boldsymbol{J}$
Initialize: K clusters $\boldsymbol{C}_k$
Distributions of textual features on each cluster: $P(t|\boldsymbol{C}_k)$
Distributions of visual features on each cluster: $P(f|\boldsymbol{C}_k)$
Distributions of images on each cluster: $P(i|\boldsymbol{C}_k)$
Distributions of images on each cluster: $P(i|\boldsymbol{C}_k)$
Posterior probabilities of K clusters for each image: $P(\boldsymbol{C}_k|i)$
Posterior probabilities of K clusters for each image: $P(\boldsymbol{C}_k|i)$
2K feature vector for each image
Adjust $\boldsymbol{C}_k$
Clusters Indexes
converged

Figure 3. The flowchart of the proposed algorithm

In Fig. 3, as discussed before, we need to initialize the images into the clusters $\boldsymbol{C}_k^{(0)}$. Here, k means the k-th cluster and the number in the superscript denotes the iterating number.

The details of the algorithm are followed:

A. Distributions on Clusters

As mentioned above, the distributions of features on different clusters could be a measurement for the images. Here comes an example. We collect 50 images in two categories, flower and people, and each image has been attached to several semantic related textual tags. Fig. 4 shows the distributions of all the textual tags on these two categories of images. We can clearly find that the features have different distributions on different categories.

The calculation of distributions could be separated into three steps. Firstly, we calculate the conditional ranking scores on the induced subgraph of each cluster; secondly, we calculate the distributions of features on the clusters; and at last, calculate the distributions of the images on the clusters with the weighted adjacency matrix and the distributions of features calculated above.

1) Ranking Functions

According to the relations between images and features, the simplest ranking of the features is to calculate the sum of weights for every feature and then calculate the proportions of the sum of weights for every feature. The formulation for calculating the ranking scores of visual features are written as follows:

$$\boldsymbol{r}_{f_j|\boldsymbol{C}_k^{(s)}} = \frac{\sum_{l\in \boldsymbol{C}_k^{(s)}} \boldsymbol{A}(l,j)}{\sum_{j=1}^{p}\sum_{l\in \boldsymbol{C}_k^{(s)}} \boldsymbol{A}(l,j)} \qquad (2)$$

Similarly, the formulation for the textual features could be written as follows:

$$\boldsymbol{r}_{t_j|\boldsymbol{C}_k^{(s)}} = \frac{\sum_{l\in \boldsymbol{C}_k^{(s)}} \boldsymbol{B}(l,j)}{\sum_{j=1}^{n}\sum_{l\in \boldsymbol{C}_k^{(s)}} \boldsymbol{B}(l,j)} \qquad (3)$$

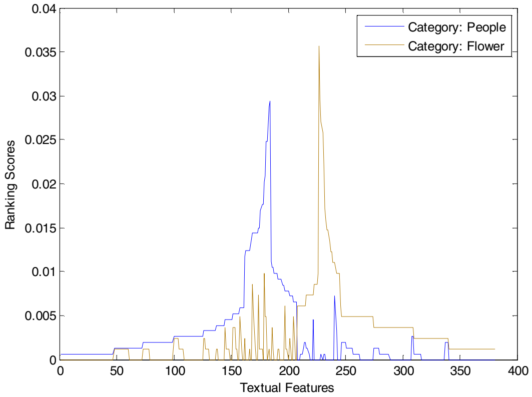


Figure 4. The distributions of all the textual tags on the flower and people categories: The brown and blue line represents the distribution of the flower and the people category, respectively.

2) *Distributions of Features on Clusters*

The relations between the textual features cannot be described in the ranking process. For example, on a subgraph for the specific cluster $\{i_1, i_2\}$ shown in Fig. 5, vertexes t_3, t_4, t_5 have no edges linked to any vertex that represent the images in the graph. So their probabilities of the distribution on this cluster should be zeros. But if we perform the ranking on the whole subgraph directly, these vertexes would receive the same ranking scores as t_1 and t_2. It would be negative for the clustering. So the direct ranking over the whole graph cannot work and we need some other ways to take advantage of the intra-relations among textual features.

Here we propose a new method to take advantage of the relations among the different textual features. We can calculate the probabilities by multiplying the ranking scores and adjacency matrix $\boldsymbol{C}$ and then make them normalized.

$$P(t_j \mid \boldsymbol{C}_k^{(s)}) = \frac{\boldsymbol{r}_{t|\boldsymbol{C}_k^{(s)}} \boldsymbol{C}}{| \boldsymbol{r}_{t|\boldsymbol{C}_k^{(s)}} \boldsymbol{C} |} \tag{4}$$

If the weight between different textual features is larger, the textual features may be more similar. And the similar features usually have similar ranking scores on the graph. So a ranking scoring may be enhanced by its similar textual features which have high ranking scores, or be decreased if its similar textual features have low ranking scores. In the practical cases, there are always some “star” textual features which have gained much more significance than other textual features. For example, in the “people” category, “face” would be a “star” textual feature as it could appear in most of surrounding texts of Web images. The close relations with “star” textual feature would help the textual feature to obtain a higher probability in distribution on that cluster. And it would help for accelerating convergence of clustering and improving the clustering performance for the images which have similar textual features.

For the visual features, intra-relations among different visual features are usually quite unapparent. So we ignore the intra-relations and simply regarded the calculated conditional ranking scores regarded as the distributions on the clusters. And in some cases, if there is no intra-relation among the different textual features, the conditional ranking scores of textual features can be regarded as the distributions of the textual features.

3) *Distributions of Images on Clusters*

After the distributions of all the features calculated, we could calculate the conditional probability distribution of images on each cluster. If the features get higher probabilities on the specific cluster, the related images would get higher probabilities on this specific cluster. And otherwise, if the features get lower probabilities on this cluster, the related images would also get lower probabilities. So for the visual features, the formulation would be simply written as

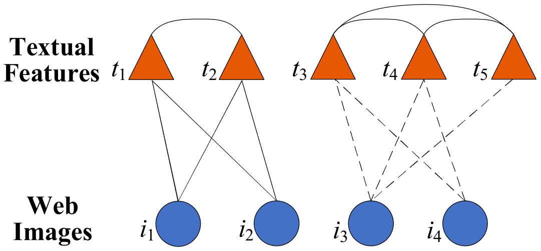


Figure 5. An example of ranking the subgraph with intra-relations

$$P(i_l \mid \boldsymbol{C}_k^{(s)}) = \frac{\sum_{j=1}^{p} \boldsymbol{A}(l,j) P(f_j \mid \boldsymbol{C}_k^{(s)})}{\sum_{i=1}^{m} \sum_{j=1}^{p} \boldsymbol{A}(l,j) P(f_j \mid \boldsymbol{C}_k^{(s)})} \tag{5}$$

Similarly, the formulation for the textual features could be written as follows:

$$P(i_l \mid \boldsymbol{C}_k^{(s)}) = \frac{\sum_{j=1}^{n} \boldsymbol{B}(l,j) P(t_j \mid \boldsymbol{C}_k^{(s)})}{\sum_{i=1}^{m} \sum_{j=1}^{n} \boldsymbol{B}(l,j) P(t_j \mid \boldsymbol{C}_k^{(s)})} \tag{6}$$

B. *Mixture Models*

As illustrated above, we build the mixture models to generate the new features for each image. For the textual features, the mixture model can be described as follows:

$$P(t_j \mid i_l) = \textstyle\sum_{k=1}^{K} \pi_{l,k} P(t_j \mid \boldsymbol{C}_k^{(s)}), \text{and} \sum_{k=1}^{K} \pi_{l,k} = 1 \tag{7}$$

Similarly, for the visual features, the mixture model can be described as follows:

$$P(f_j \mid i_l) = \textstyle\sum_{k=1}^{K} \theta_{l,k} P(f_j \mid \boldsymbol{C}_k^{(s)}), \text{and} \sum_{k=1}^{K} \theta_{l,k} = 1 \tag{8}$$

This mixture model can be considered as describing the probabilities to generate a link between the j-th textual feature t_j and the l-th image i_l. And the component coefficients $\pi_{l,k}$ can be regarded as the posterior probability $P(\boldsymbol{C}_k^{(t)} \mid i_l)$. So the component coefficients can be used as the new feature to adjust the current clusters. Since every image would have two mixture models which are generated by textual features and visual features separately, the image’s new feature would be combined by both component coefficients.

Now the problem becomes to estimate the component coefficients. Here we use the mixture model by textual features as an example, the procedure would be exactly same for the mixture models by visual feature. We adopt the EM algorithm to obtain the best estimation for the coefficients.

Let $\boldsymbol{\Theta}$ be the m by K matrix, consisting of the K-dimensional parameter vector $\pi_{l,k}$ for the mixture model. Here, the objective function can be described as

$$L(\boldsymbol{\Theta}) = P(\boldsymbol{B} \mid \boldsymbol{\Theta}) = \prod_{l}^{m} \prod_{j}^{n} P(i_l, t_j \mid \boldsymbol{\Theta})^{\boldsymbol{B}(l,j)} \tag{9}$$

And in the E-step:

$$\begin{aligned} & P(z = k \mid t_j, i_l, \boldsymbol{\Theta}^0) \\ & \propto P(t_j, i_l \mid z = k) P(z = k \mid \boldsymbol{\Theta}^0) \\ & = P(t_j \mid z = k) P(i_l \mid z = k) P^0(z = k) \end{aligned} \tag{10}$$

Here, z denotes the index of cluster. So $P(t_j|z=k)$ is equal to $P(t_j|\boldsymbol{C}_k)$.

Then in the M-step:

$$P(z=k)=\frac{\sum_{l=1}^{m}\sum_{j=1}^{n}B(l,j)P(z=k\mid t_j,i_l,\boldsymbol{\Theta}^0)}{\sum_{l=1}^{m}\sum_{j=1}^{n}B(l,j)} \tag{11}$$

By setting $\boldsymbol{\Theta}^0 = \boldsymbol{\Theta}$, the whole process can be repeated. At each iteration, we use (10) and (11) to update and finally $\boldsymbol{\Theta}$ will converge to a local maximum. Each parameter vector $\pi_{l,k}$ can be calculated using Bayesian Rules:

$$\pi_{l,k}=P(z=k\mid i_l)=\frac{P(i_l\mid z=k)}{\sum_{j=1}^{K}P(i_l\mid z=j)P(z=k)} \tag{12}$$

C. Clustering Adjusting

After the features calculated, we could use the new feature to adjust the clusters to get better cluster results.

Here we adopt the K-means algorithm to accomplish the task. We donate the new feature vector ($\pi_{l,1}$, $\pi_{l,2}$, ... , $\pi_{l,k}$) as $\boldsymbol{d}_{i_l}$. The centers for each cluster can thus be calculated as follows, which is the mean for all items in each cluster:

$$\boldsymbol{d}'_k=\frac{\sum_{i_l\in \boldsymbol{C}_k^{(s)}}\boldsymbol{d}_{i_l}}{|\boldsymbol{C}_k^{(s)}|} \tag{13}$$

Here, $|\boldsymbol{C}_k^{(s)}|$ denotes the size of the cluster k. Next, the distance between an image and cluster center ($D(i_l,\boldsymbol{C}_k^{(s)})$) can be defined by 1 minus cosine similarity:

$$D(i_l,\boldsymbol{C}_k^{(s)})=1-\frac{\boldsymbol{d}_{i_j}\bullet\boldsymbol{d}'_k}{|\boldsymbol{d}_{i_j}\|\boldsymbol{d}'_k|} \tag{14}$$

Here, $|\boldsymbol{d}_k|$ and $|\boldsymbol{d}'_k|$ denotes the modulus of $\boldsymbol{d}_k$ and $\boldsymbol{d}'_k$. Then we can adjust the clusters by their distances.

D. Algorithm Summary

As illustrated above, the proposed algorithm is summarized in Fig. 6.

For an algorithm, efficiency is always an important evaluation of performance, especially when it runs on the big data. In this work, the time complexity of proposed algorithm is comprised of three parts: ranking and distributions calculating, mixture model estimation, and clustering adjustment. For the ranking and distributions calculating part, the time complexity is $O(m(n+p)+n^2)$. Particularly, when the number of images m is quite large, the number of textual tags n would be much smaller than m, and the complexity may be reduced to $O(m(n+p))$. For the mixture model estimation part, the time complexity is $O(t_1Km(n+p))$ where the t_1 denotes the iteration number of EM algorithm. And for the clustering adjusting part, the time complexity should be $O(mK^2)$. So for the overall, the time complexity is $O(t_2\ (n^2 + t_1Km\ (n + p) + mK^2))$, and the t_2 denotes the overall iteration number.

V. Experiments

We perform the proposed algorithm on two image datasets which are selected from the public Web image datasets, MIR-Flickr25K [13] and IAPR TC-12 Benchmark [14], respectively.

```
Algorithm: Web Image Clustering
Input: The Graph Model $\boldsymbol{G} = \{\boldsymbol{V}, \boldsymbol{E}\}$
       The Adjacency Matrix $\boldsymbol{J}$
       Cluster Number $K$
Procedure:
Initialize: $s = 0$;
    initial clusters $\boldsymbol{C}_k^{(s)}$
Iterations:
    for each type of feature:
        calculating the conditional ranking scores;
        calculating the distribution: $P(t_j|\boldsymbol{C}_k^{(s)}), P(f_l|\boldsymbol{C}_k^{(s)})$ ;
        calculating $P(i|\boldsymbol{C}_k^{(s)})$
          estimating coefficients of the mixture model
          $P(t_j \mid i_l) = \sum_{k=1}^{K}\pi_{l,k}P(t_j \mid \boldsymbol{C}_k^{(s)})$
    get the 2$K$-dimensional new feature vector
    adjust the clusters:
    if converged, then end the procedure; otherwise,
    $s = s + 1$
Output: Cluster index for every image
```

Figure 6. The proposed algorithm

A. Data Preparation

1) Dataset 1

The MIR-Flickr25K dataset is collected by Huiskes from the Flickr website[3] in 2008. It consists of 25000 images which were downloaded from Flickr through its public API. For each image, it has a description tag text file, a camera information file and a copyright license file. In practice, we manually select two categories of images including flowers and people. The size of the categories is listed in the Table I.

We use the existing tags in the tag text file as the textual features of the images. The weighted matrix $\boldsymbol{B}$ can be defined as follows:

$$\boldsymbol{B}(l,j)=\begin{cases}1, & \text{if } l\text{-th image has the } j\text{-th textual feature}\\ 0, & \text{otherwise}\end{cases} \tag{15}$$

And calculate the weights of intra-relations among the tags as follows:

$$\boldsymbol{C}(i,j)=\begin{cases}1, & \text{if } i=j\\ \dfrac{num(i,j)}{m}, & \text{if } i\neq j\end{cases} \tag{16}$$

Here, $num(i, j)$ denotes the number of images which contain both the textual feature $t_i$and t_j.

TABLE I. The category size of the dataset

Dataset	Category Name	Category Size
MIR-Flickr 25K	Flower	444
	People	329
IAPR TC-12 Benchmark	Architecture	137
	Mountain	141

[3] http://www.flickr.com

For the visual features, we use the SIFT image descriptors [15]. To be specific, we firstly gray the images and normalize into an image size of 64*64, then extract 81 patches from each image and at last generate the adjacency matrix ***A***.

2) *Dataset 2*

The image collection of the IAPR TC-12 Benchmark [14] consists of 20,000 still natural images, including pictures of different sports and actions, photographs of people, animals, cities, landscapes and many other aspects of contemporary life. Each image is associated with a text annotation which includes the image title, description, location, date, etc. Here, we also manually select two categories including mountains and architectures.

The textual features are generated from mining the image titles and descriptions in the image annotation files. The visual features are represented by SIFT image descriptors as well.

For Dataset 2, the adjacency matrix ***A*** and ***C*** are generated in the same way like for Dataset 1 while matrix ***B*** is generated by the traditional tf-idf method. Here comes the formulation:

$$B(l,j)=\sum_{l=1}^{m} freq(t_j,i_l)\log\frac{m}{num(t_j)} \tag{17}$$

Here, m denotes the size of the image dataset, $freq(t_j,i_l)$ denotes the frequency of the textual feature t_j in the surrounding text of image i_l; and $num(t_j)$ denotes the number of images which contains the textual feature t_j.

B. *Experiments Setting*

Firstly, we perform our proposed algorithm with the two dataset. And next, we compare the results with some other algorithms like K-means, RankClus [6] and CIHC [10]. With these comparisons, we can see the superiority of our proposed algorithm. Then we perform some experiments to verify the effectiveness of our proposed distribution calculating method.

We take the Normalized Mutual Information (NMI) measure to evaluate the accuracy of clustering results. Suppose that we have N objects, K clusters, and two clustering results. Then we define a function $n(i, j)$, where the cluster labels i, j = 1, 2, ..., K, be the number of objects that are labelled i in the first clustering result (e.g., generated by the algorithm) and labelled j in the second clustering result (e.g., the ground truth). Then, we define three distributions of cluster labels: the joint distribution $P(i,j)=\frac{n(i,j)}{N}$, row distribution $P_1(j)=\sum_{i=1}^{K}P(i,j)$ and column distribution $P_2(i)=\sum_{j=1}^{K}P(i,j)$, and then NMI can be defined like this:

$$NMI=\frac{\sum_{i=1}^{K}\sum_{j=1}^{K}P(i,j)\log(\frac{P(i,j)}{P_1(i)P_2(j)})}{\sqrt{\sum_{j=1}^{K}P_i(j)\log(P_1(j))\sum_{i=1}^{K}P_2(j)\log(P_2(i))}} \tag{18}$$

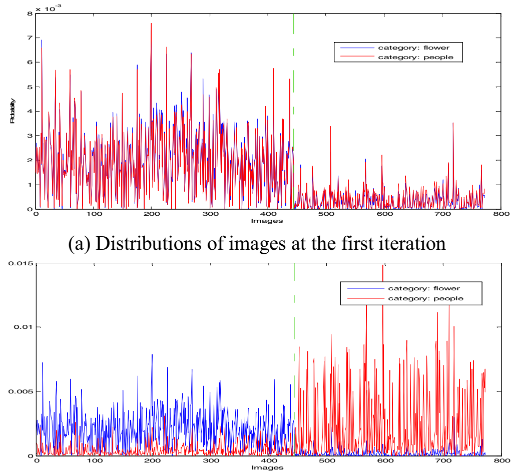
C. *Experiments Results*

Firstly, we perform our proposed algorithm on the two datasets. Here, we list some intermediate results during the iteration process and some representative images of clustering results of Dataset 1 in Fig. 7 and Fig. 8.

The Fig. 7 shows the probability distributions of images on two clusters, which are calculated according to the distributions of textual features. The top figure shows distributions at the beginning while the bottom one shows at the end of iteration. In the figures, the blue curve represents the probabilities on the *flower* category, the red curve represents the probabilities on the *people* category, and the green vertical dotted line separates the categories. We can see that in the beginning, the distributions on both categories are quite similar (Fig. 7(a)). And eventually at the last round of iteration, the distributions of images are clearly distinguishable (Fig. 7(b)). In the Fig. 8, it shows some representative images of the cluster results. We can see that the images in the same category have various visual features. And it is the diversity of images in the same category that leads to the low NMI accuracy.

The results of different clustering algorithm are listed in Table II. Here, we noted our proposed algorithm as Web Images Clustering with Integrating Visual and Textual Features (WIC-IVTF).

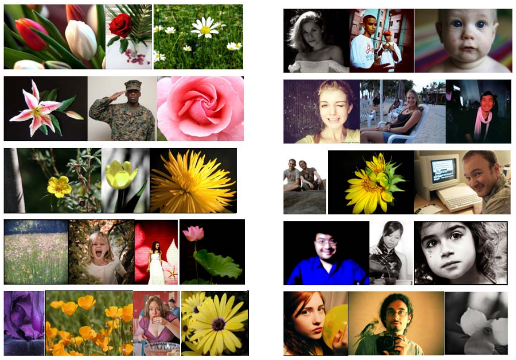
The K-means algorithm only takes the visual features as input. The comparison between our proposed algorithm and the K-means aims to verify the superior of integrating both visual and textual features to only using visual features. Obviously, the results show that our algorithm has much better



(a) Distributions of images at the first iteration

(b) Distributions of images at the last iteration

Figure 7. Distributions during the iterations



(a) Cluster label: 1

(b) Cluster label: 2

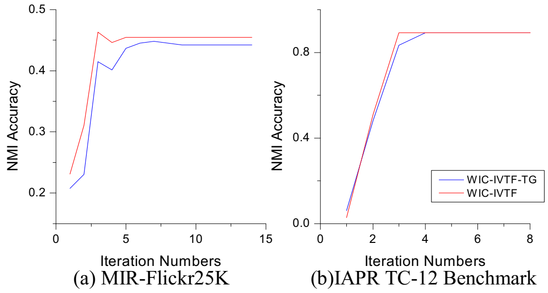
Figure 8. Clustering results on Dataset 1

TABLE II CLUSTERING RESULTS

Algorithms	NMI Accuracy on MIR-Flickr25K	NMI Accuracy on IAPR TC-12
WIC-IVTF	**45.46%**	**89.13%**
K-means	28.01%	65.80%
RankClus	44.65%	83.37%
CIHC	43.53%	85.15%

TABLE III CLUSTERING RESULTS FOR COMPARISON BETWEEN WIC-IVTF AND WIC-IVTF-TG

Algorithms	NMI Accuracy on MIR-Flickr25K	NMI Accuracy on IAPR TC-12
WIC-IVTF	**45.46%**	**89.13%**
WIC-IVTF-TG	44.23%	**89.13%**



(a) MIR-Flickr25K (b)IAPR TC-12 Benchmark

Figure 9. NMI accuracy during iterations

performance, especially on Dataset 1. Since the images in the same category of Dataset 1 have various visual features, the K-means algorithms cannot obtain satisfied clustering results. But, the unsatisfied results also reveal the effectiveness and robustness of our proposed algorithm.

Similarly, to verify the superior of integrating both visual and textual features to only using textual features, we compare our proposed algorithm with RankClus algorithm. The clustering results listed in Table II shows our proposed algorithm has obtained higher NMI accuracy.

And next, we compare WIC-IVTF with the CHIC algorithm which is one of the state-of-art algorithms for Web images clustering simultaneously using visual and textual features. As listed in Table II, our method has obtained better accuracies on both datasets. Specially, the performance on Dataset 1 has been improved by about 2% while the performance on Dataset 2 has been improved by about 4%.

Next, we perform some comparison experiments to prove that our method for calculating the distributions on the clusters is helpful. We replace the real adjacency matrix $\boldsymbol{C}$ with the unity matrix $\boldsymbol{E}$ and the graph model is changed into a tripartite graph. So it is noted as Web Image Clustering with Integrating Visual and Textual Feature - Tripartite Graph (WIC-IVTF-TG). The comparison results are listed in Table III and Fig. 9.

From the results listed in the Table III, we can see that our proposed distribution calculation method has enhanced the clustering results on the MIR-Flickr25K dataset. But it has not improved the NMI accuracy. With analyzed the details of the clustering results, we find that the indexes of both results are converged at same arrangements and only 4 images are not correctly clustered (2 mountain images are clustered with other 135 architecture images while 2 architecture images are clustered with other 139 mountain images). Some of visual and textual features of the 4 images are quite ambiguous and the clustering results are the optimal results for clustering methods.

On the other side, Fig. 9 shows the convergent process of the clustering. The left chart (Fig. 9(a)) shows that at the fifth iteration, we could get the results converged while the WIC-IVTF-TG needs about 4 more iterations. And the right chart (Fig. 9(b)) shows that using the WIC-IVTF method can get the clustering result converged faster than using WIC-IVTF-TG as well. Since the time complexity of our calculating method is far less than one round of iteration, the efficiency of clustering has been certainly improved by our method. Furthermore, comparing the two charts in Fig. 9, we can also find that the efficiency of our proposed calculation would be magnified when the size of dataset is larger. So with both Table III and Fig. 9, we can get a conclusion that our proposed calculation method can enhance the NMI accuracy and improve the efficiency of the clustering method, especially for the large dataset.

VI. CONCLUSIONS

In this paper, two graph models are built by making use of visual and textual features. A new iterative algorithm for Web image clustering is proposed. Intensive experiments have been carried out to evaluate the performance of the proposed method on MIR-Flickr25K and IAPR TC-12 Benchmark dataset. Experimental results show that the proposed method is able to achieve a clustering accuracy about 45% with 773 images from MIR-Flickr25K and 89% with 278 images from IAPR TC-12 Benchmark, which outperform the compared method.

ACKNOWLEGEMENT

This work is supported by Shenzhen Science Fundamental Research Program (No. JCYJ20130329175141512)

REFERENCES

[1] F. Fauzi and M. Belkhatir, "Image understanding and the web: a state-of-the-art review," Journal of Intelligent Information Systems, vol. 43, pp. 271-306, 2014.

[2] S. Gordon, H. Greenspan, and J. Goldberger, "Applying the information bottleneck principle to unsupervised clustering of discrete and continuous image representations," in Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, 2003, pp. 370-377.

[3] Y. Yang, D. Xu, F. Nie, S. Yan, and Y. Zhuang, "Image clustering using local discriminant models and global integration," Image Processing, IEEE Transactions on, vol. 19, pp. 2761-2773, 2010.

[4] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen, "Hierarchical clustering of WWW image search results using visual, textual and link information," in Proceedings of the 12th annual ACM international conference on Multimedia, 2004, pp. 952-959.

[5] B. Gao, T.-Y. Liu, T. Qin, X. Zheng, Q.-S. Cheng, and W.-Y. Ma, "Web image clustering by consistent utilization of visual features and surrounding texts," in Proceedings of the 13th annual ACM international conference on Multimedia, 2005, pp. 112-121.

[6] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu, "Rankclus: integrating clustering with ranking for heterogeneous information network analysis," in Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, 2009, pp. 565-576.

[7] Y. Yan, G. Liu, S. Wang, J. Zhang, and K. Zheng, "Graph-based clustering and ranking for diversified image search," Multimedia Systems, pp. 1-12, 2014.

[8] P. Xie and E. Xing, "Integrating Image Clustering and Codebook Learning," 2015.

[9] F. Wu, H.-T. Pai, Y.-F. Yan, and J. Chuang, "Clustering results of image searches by annotations and visual features," Telematics and Informatics, vol. 31, pp. 477-491, 2014.

[10] M. Rege, M. Dong, and J. Hua, "Graph theoretical framework for simultaneously integrating visual and textual features for efficient web image clustering," in Proceedings of the 17th international conference on World Wide Web, 2008, pp. 317-326.

[11] D. Zhou and C. J. Burges, "Spectral clustering and transductive learning with multiple views," in Proceedings of the 24th international conference on Machine learning, 2007, pp. 1159-1166.

[12] F. Wu, Y.-H. Han, and Y.-T. Zhuang, "Multiple hypergraph clustering of web images by miningword2image correlations," Journal of Computer Science and Technology, vol. 25, pp. 750-760, 2010.

[13] M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in Proceedings of the 1st ACM international conference on Multimedia information retrieval, 2008, pp. 39-43.

[14] M. Grubinger, P. Clough, H. Müller, and T. Deselaers, "The iapr tc-12 benchmark: A new evaluation resource for visual information systems," in International Workshop OntoImage, 2006, pp. 13-23.

[15] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009, pp. 1794-1801.