Clustering and Ranking of Image Search Results

© Elena Sivogolovko *

University of Saint-Petersburg
efecca@gmail.com

Abstract

In a typical content-based image retrieval (CBIR) system, query result is a set of images sorted by feature similarities with respect to the query. We introduce a new approach to CBIR result representation. We propose that CBIR system should retrieve image clusters, which elements should be sorted by the most meaningful feature similarities. Actually, this paper does not present a full approach but rather work in progress.

1 Introduction

In image search, good organization of the search results is as important as the search accuracy. Many existing CBIR search engines return large quantity of search results, ranked by their relevance to the given query. Users have to go through the list and look for the desired ones. This is a time consuming task since the returned results usually contain multiple topics and these topics are mixed together. Things become even worse when one topic is dominating but it is not what the user desires. A possible solution to this problem is to cluster search results into different semantic groups. In traditional CBIR area, image clustering techniques are often used to design a convenient user interface, which helps to make more meaningful representations of search results. However, as the images were usually represented by low level visual features, it is hard to get a good clustering result from semantic perspective, because images with high feature similarities to each other may be very different in terms of semantics. This is known as the semantic gap problem.

In this paper, we consider the problem of clustering and ranking image search results. We propose a framework to represent an image query result as a clustering structure where elements in each cluster are sorted by the most meaningful features.

2 Related work

In this section we will review graph partitioning and spectral clustering method, which are the foundation of our framework. Such approaches are successfuly used in other CBIR systems, for example F-I-T[6] and ImageRank[7], but these systems do not consider information about image features meaning, which can be obtained during clustering process. However, we suppose that such kind of information can be usefull for search result representation.

2.1 Bipartite graph model

Graph model is widely used in different clustering tasks, such as sparse matrix partitioning, circuit partitioning, image segmentation and document clustering. It can also be used in image clustering. Qiu[4] used the undirected bipartite graph(Figure 1) to represent the relationship between images and their low-level features. This graph is constructed as following. Assuming each image in the database is represented by an m-bin colour histogram and $H_k = (h(c_1), h(c_2) \dots h(c_m))$ denotes the histogram of the k-th image, where $k = 1, 2, \dots n$ and the value of $h_k(c_i)$ indicates an association of the colour c_i with the k-th image. Then the bipartite graph can be represented by a triplet $G = (I, C, W)$, where $I = \{i_1, i_2, \cdots, i_n\}$ is a set of images, $C = \{c_1, c_2, \cdots, c_m\}$ is a set of colors and W is a set of edges connecting vertices from different vertex sets, i.e., $W = \{< i, j > | i \in I, j \in C\}$. The weight of edge $W_{ij} = h_i(c_j)$ is the j-th colour bin count of the i-th image.

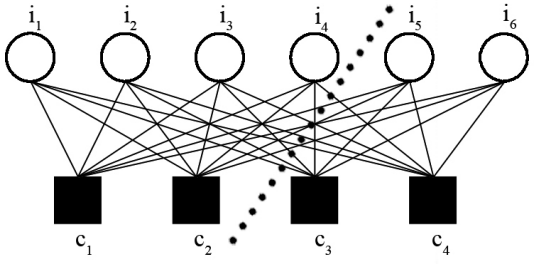

Figure 1: Images and features bipartite graph. Squares and circles represent low-level features(colors in our case) and images respectively.

Note that we can consider this graph as a similarity graph with similarity function $S = W_{ij}$. The adjacency matrix of G will be written as:

$$M = \begin{array}{c} \\ I \\ C \end{array} \begin{array}{ccc} I & C \\ 0 & W \\ W^T & 0 \end{array} \quad (1)$$

In this context co-clustering of images and features can be translated to a graph partitioning problem.

* This work is partially supported by Russian Foundation for Basic Research under grant 07-07-00268.

Proceedings of the Spring Young Researcher's Colloquium on Database and Information Systems, Moscow, Russia, 2009

2.2 Graph partitioning and spectral co-clustering

The idea of clustering is to separate a set of points into different groups according to their similarities. For data given in a form of a similarity graph this problem can be restated as follows: we want to find a partition of the graph such that the edges between different groups have very low weight (which means that points in different clusters are dissimilar from each other) and the edges within a group have high weight (which means that points within the same cluster are similar to each other). Given a similarity graph $G =< V, E >$ (in our case $V = I \cup C$ and $E = W$) with adjacency matrix M, the simplest and most direct way to construct a partition is to solve the mincut problem. This consists of choosing the partition $\{A_1, \ldots, A_k\}$ which minimizes

$$cut(A_1, \ldots A_k) = \sum_{i=1}^{k} cut(A_i, \bar{A}_i) \tag{2}$$

where

$$cut(A_i, \bar{A}_i) = \sum_{i \in A, j \in \bar{A}_i} m_{ij}, \tag{3}$$

m_{ij} is element of adjacency matrix M, which corresponds to edge form i vertex to j vertex, and $\bar{A}_i$ is the complement of a subset $A \in V$. The problem is that in many cases, the solution of mincut simply consists in separating one individual vertex from the rest of the graph. Of course this is not what we want to achieve in clustering, as clusters should be reasonably large groups of points. One way to circumvent this problem is to explicitly request that the sets $A1, \ldots, Ak$ are "reasonably large". The two most common objective functions which encode this are RatioCut and the normalized cut – Ncut. In RatioCut, the size of a subset A of a graph is measured by its number of vertices $|A|$, while in Ncut the size is measured by the weights of its edges $vol(A)$.

$$RatioCut(A_1, \ldots A_k) = \frac{\sum_{i=1}^{k} cut(A_i, \bar{A}_i)}{|A_i|} \tag{4}$$

$$Ncut(A_1, \ldots A_k) = \frac{\sum_{i=1}^{k} cut(A_i, \bar{A}_i)}{vol(A_i)} \tag{5}$$

In this work we use Ncut as objective function, because it implements both clustering tasks mentioned below: Ncut minimizes the between-cluster similarity and maximizes the within-cluster similarity unlike RatioCut, that implements only first of them. Ncut could be rewritten as a trace minimization problem and leads to retrieving first k generalized eigenvectors of

$$Lv = \lambda Dv \tag{6}$$

here D is a diagonal matrix with $D_{ii} = \sum_{k=1}^{n} m_{ik}$, which is also called degree matrix, and $L = D - M$ is unnormalized graph Laplacian matrix. After that, the desired image clusters can be obtained by running some routine clustering algorithms such as k-means on these eigenvectors.

3 Co-clustering and ranking framework

Suppose the dashed line in Figure 1 shows the very partition that minimizes two part cut. After applying clustering algorithm, we will obtain two subsets $\{i_1, i_2, i_3, i_4, c_1, c_2\}$ and $\{i_5, i_6, c_3, c_4\}$. We define such clusters as mixed. Consequently, the low-level features are divided into two feature clusters $\{c_1, c_2\}$ and $\{c_3, c_4\}$, while the images are divided into two image clusters $\{i_1, i_2, i_3, i_4\}$ and $\{i_5, i_6\}$ respectively. The most of CBIR systems return obtained image subsets and do not use feature subsets at all. We propose to sort images in each image cluster using low-level features from corresponding feature cluster. We suggest that such kind of ranking helps us to improve image retrieval results.

To sort images by more than one feature we need a data fusion method. In general data fusion is use of techniques that combine data from multiple sources and gather that information in order to achieve inferences, which will be more efficient and potentially more accurate than if they were achieved by means of a single source. In our case source is a list of images ranked by one of the selected low-level features. A lot of commonly used data fusion algorithms such as CombSUM, CombMNZ are successfuly used in text retrieval. CombMNZ is considered to outperform other data fusion algorithms. It works as follows. Element in the result ranked-list gets rank equaled to the sum of all its ranks in fused lists multipied by the number of lists in which this element exists with non-zero rank:

$$rank_{result}(i) = \sum_{f_{usedlists}} rank_{list}(i) * nz \tag{7}$$

where nz is number of non-zero rank lists. To summarize, our algorithm of images and features co-clustering and ranking can be listed as below.

Algorithm 1 Co-clustering and ranking algorithm

Require: Image set I and the vectors H representing histograms of these images

1: Form a bipartite graph $G =< I, C, W >$
2: Compute matrix M, $D : D_{ii} = \sum_{k=1}^{n} m_{ik}$ and $L = D - M$.
3: Compute the first k eigenvectors $v1, \ldots, v_k$ of the generalized eigenproblem $Lv = \lambda Dv$.
4: Let $V \in R^{n \times k}$ be the matrix containing the vectors $v_1, \ldots, v_k$ as columns.
5: For $i = 1 \ldots n$ let $y_i \in R^k$ be the vector corresponding to the i-th row of V.
6: Cluster the points $(y_i)_{i=1,\ldots,n}$ in R^k with the k-means algorithm into mixed clusters $C_1, \ldots, C_k$.
7: Extract image clusters from mixed ones. Rank images in every cluster use features from corresponding feature clusters and data fusion method CombMNZ.
8: Return set of image clusters $I_1, \ldots, I_k$.

It is important to note that we should normalize feature values before applying the algorithm to the graph. In fact, most widely used image features such as color histogram, CPAM histogram and Spatial CPAM, have already introduced the mechanism of normalization.

4 Future Work

First, we will test our framework on real image queries. We plan to use two different databases from CorelPhoto-Set collection: ImageDBCorel database(650 images in 9 semantic classes) and CorelSmall-100 database(100 images in 16 semantic classes). Second, if our co-clustering and ranking scheme gives good results, we use the following research approach.

As in many other clustering systems, the most difficult question in our co-clustering scheme is "How will we define a number of clusters?". In our case such clustering algorithms as G-means and X-means, which provide a number of clusters, are not applicable because we need this number for eigenvectors search before clustering algorithm starts. There is some research in spectral clustering, where authors propose to use eigengap in order to define a correct number of clusters, so this method can be tested in our framework too.

Concerning directly clustering algorithm, K-means is easy-to-implement classical method which can provide a good clustering scheme. In our case its main disadvantage is the assumption that image clusters are crisp, but it is easy to see that in most cases image clusters are overlaping. Therefore we suppose that fuzzy clustering methods(for example fuzzy K-means) can provide more realevant result in image clustering than crisp ones and we will check this hypothesis.

Also our framework should be extended for using multiple feature sets: color histograms and texture features simultaneously, because only color data is insufficient for retrieval in collections of heterogeneous images. Actually, in this case we have two important problems. First of them is definition a set of features which should be used for clustering. A wide variety of color, texture and shape features have been proposed to represent image content, so selecting the most meaningful ones is really difficult. The second problem is a normalization problem. If we combine all our image features values into one image representation vector, this vector should be normalized. Suitable normalization method is needed.

5 Conclusion

This research is conducted under the supervision of Boris Novikov. The contribution of this paper is a model of new co-clustering and ranking scheme. This scheme combines different information retrieval technics, such as spectral clustering, ranking and data fusion methods, in order to improve image retrieval effectiveness. We suppose that our search results representation should be realy convinient and helpful. We plan to include the developed framework into CBIR system Foto Finder, which is being created in our research group.

References

[1] Reginald E. Hammah and John H. Curran: A Tutorial on Spectral Clustering. Technical Report No. TR-149, August 2006

[2] Ilya Markov, Natalia Vassilieva: Building Up Low-Level Centroids for Groups of Perceptually Similar Images.

[3] E.A. Fox, J.A. Shaw. Combination of multiple searches. In TREC 2 (1994), 243249.

[4] Guoping Qiu: CBipartite Graph Partitioning and Content-based Image Clustering.

[5] Manjeet Rege, Ming Dong, Jing Hua: Clustering Web Images with Multi-modal Features. SIGMM07, September 2328, 2007.

[6] Bin Gao, Tie-Yan Liu, Tao Qin, Xin Zheng1, Qian-Sheng Cheng, Wei-Ying Ma: Web Image Clustering by Consistent Utilization of Visual Features and Surrounding Texts. SIGMM05, November 6-11, 2005.

[7] Xiaofei He, Wei-Ying Ma, Hongjiang Zhang: ImageRank : Spectral Tecnhiques For Structural Analysis of Image Database.