Image and Feature Co-clustering

Guoping Qiu
School of Computer Science, The University of Nottingham
qiu@cs.nott.ac.uk

Abstract

The visual appearance of an image is closely associated with its low-level features. Identifying the set of features that best characterizes the image is useful for tasks such as content-based image indexing and retrieval. In this paper, we present a method which simultaneously models and clusters large sets of images and their low-level visual features. A computational energy function suited for co-clustering images and their features is first constructed and a Hopfield model based stochastic algorithm is then developed for its optimization. We apply the method to cluster digital color photographs and present results to demonstrate its usefulness and effectiveness.

1 Introduction

In this paper, we propose to use a bipartite graph to model the images and their contents simultaneously. For the two sets of vertices of a bipartite graph, we associate one set with image content descriptors and the other with the images. The edges linking these two sets of vertices measure the degrees of association between the content descriptors and the images. We then introduce graph partitioning to cut the graph such that images and the features that are most strongly associated with each other are clustered into the same group. Each group of images formed in this way manifests a certain visual theme which is strongly linked to its associated visual features, which in turn provides the content semantics for the images, thus facilitating image indexing and retrieval.

Graph partitioning has many well-established theoretical results, which makes the approach very attractive. Recent literature includes in the areas of image segmentation [2, 3] and document clustering [4, 5]. Although graph partition problem is NP-complete, which will make it computationally unattractive, recent results have shown that graph partition can be efficiently computed by using spectral algorithms [2-5]. Bipartite graph can also be partitioned by a Hopfield network [6]. In this work, we develop a Hopfield network based stochastic solution to bipartite graph partitioning for the co-clustering of images and their features.

The organization of the paper is as follows. In section 2, we first briefly describe the appearance indexing scheme [8] for image content representation and then introduce the use of a bipartite graph to simultaneously model images and their content descriptors. Section 3 presents a Hopfield neural network based approach to the partitioning of bipartite graph for the co-clustering of features and images. Section 4 presents experimental results and section 5 concludes the paper.

2 Simultaneous modeling images and their features

2.1 Appearance Indexing

Appearance indexing [7, 8] first de-composes a given image into multilevel Gaussian pyramid. At each level, the image is represented in an opponent color space. At each level, image patches (blocks) of m x n pixels, are formed. For each block, we form two appearance vectors, A_l the achromatic appearance vector, and C_l the chromatic appearance vector. Appearance prototypes (VQ codebook) are created using a simple unsupervised neural network. We use 4 x 4 pixel patches for its moderate computational complexity, and which will cover areas of 4 x 4, 8 x 8, 16 x 16, 32 x 32, … in the original image. We have used over 15 million patches of 4 x 4 pixels obtained from natural color images to train the appearance prototypes. Figure 1 shows examples of achromatic and chromatic appearance prototypes used in this paper. It is important to note that these appearance patterns will be used at various levels of the pyramid, therefore, we have equivalent appearance prototype patterns of multiple resolutions. With these prototypes, appearance indexing can be formed. In this paper, we form an achromatic appearance histogram HA by indexing the achromatic appearance prototypes and a chromatic appearance histogram HC by indexing the chromatic appearance prototypes at 3 levels of the Gaussian pyramid. We then concatenate HA and HC of all levels to form the overall image descriptor (a 384-d vector).

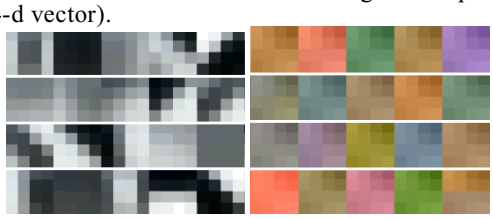

Figure 1: Examples of 4 x 4 appearance prototypes used to construct image descriptors in this paper.

2.2 Modeling Images and their appearance histograms using a bipartite graph

Let H = {h(i)} be the appearance histogram, where h(i) is the count of occurrences of the ith appearance prototype. Each bin in the histogram is associated with certain low-level properties, see Figure 1. The value of h(i) indicates an association of the ith prototype with the image. A larger value indicates that there are more patterns in the image have appearance close to that prototype, and that prototype is more important in relation to the visual appearance of the image. In traditional content-based image retrieval, image similarity is based on either L_1 or L_2 norms of the histograms [7, 8]. The relative importance of each bin to an image is not explicitly identified/used. Knowing which low-level features are more strongly associated with an image can provide useful information to search for images. For example, if the bluish bins of an image's chromatic appearance histogram have large counts, then it can be reasoned that the image contains bluish color scenes. This "bluish color theme" can be regarded as a semantic term of that image, which will in turn enable semantic based image retrieval. Similarly, if the counts of the bins in the achromatic histogram corresponding to prototypes of strong spatial changes are large, then it can be reasoned that the image contains busy or high frequency areas. We will study the issues of how these semantics can be effectively used for image retrieval in another paper. In this current paper, we present a method to co-cluster appearance histograms and images.

A graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ is a bipartite graph if it consists of two classes of vertex, $\boldsymbol{X}$ and $\boldsymbol{Y}$, $\boldsymbol{V} = \boldsymbol{X} \cup \boldsymbol{Y}$, $\boldsymbol{X} \cap \boldsymbol{Y} = \varnothing$, and each edge in $\boldsymbol{E}$ has one endpoint in $\boldsymbol{X}$ and the other endpoint in $\boldsymbol{Y}$. We denote an undirected bipartite graph by the triple $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{W})$, where $\boldsymbol{X} = \{x_1, x_2, \ldots x_m\}$, $\boldsymbol{Y} = \{y_1, y_2, \ldots y_n\}$, $\boldsymbol{W} = \{w_{kl}\}$, where $w_{kl} > 0$ is the weight between vertices k and l, $w_{kl} = 0$ if there is no edge between vertices k and l.

We wish to model the appearance histograms and images of an image database simultaneously using a bipartite graph $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{W})$. Assuming each image in the database is represented by an m-bin appearance histogram, and $H_l = (h_l(1), h_l(2), \ldots, h_l(m))$, denotes the histogram of the lth image, where $l = 1, 2, \ldots n$. A straightforward mapping of the prototypes, the histograms and the images to $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{W})$ is as follows: $\boldsymbol{X} = (ai_1, ai_2, \ldots, ai_m)$, where, ai_k represents the ith appearance prototype, i.e., each vertex in $\boldsymbol{X}$ corresponds to an appearance prototype of Figure 1 (also at 3 different resolutions). $\boldsymbol{Y} = \{y_1, y_2, \ldots y_n\}$, and y_l represents the lth image in the database. $\{w_{kl}\} = \{ h_l(k) \}$, $k = 1, 2, \ldots, m$, $l = 1, 2, \ldots, n$, i.e., the weight of the edge (k, l) is the kth appearance histogram bin count of the lth image. In this way, an image database is completely characterized by the bipartite graph $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{W})$. In the next section, we present an algorithm to partition the graph such that images and their most important features are clustered simultaneously.

3 Co-clustering images and their appearance features

We consider the case of bi-partitioning the bipartite graph $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{W})$, where the vertices of $\boldsymbol{X}$ are partitioned into two sub sets $\boldsymbol{X} = \boldsymbol{X_1} \cup \boldsymbol{X_2}$, and simultaneously, the vertices of $\boldsymbol{Y}$ are also partitioned into two sub sets $\boldsymbol{Y} = \boldsymbol{Y_1} \cup \boldsymbol{Y_2}$. Graph partition is in general NP-complete. However, recent research using eigenvector or spectral methods for graph partitioning has demonstrated that the problem can be solved quite efficiently [2-5]. A key factor that affects the quality of the solution is the cut criterions used. This criterion is inevitably task dependent. In this section, we introduce a solution based on the Hopfield neural network model [6]. We first describe a computational energy function suited for the co-clustering of images and their appearance features and then describe a stochastic algorithm for optimizing the energy function.

3.1 The computational energy function

A partition divides $\boldsymbol{X} = \boldsymbol{X_1} \cup \boldsymbol{X_2}$, and $\boldsymbol{Y} = \boldsymbol{Y_1} \cup \boldsymbol{Y_2}$. Let us assume that $\boldsymbol{X}_1$ is paired with $\boldsymbol{Y}_1$ and $\boldsymbol{X}_2$ paired with $\boldsymbol{Y}_2$. In our current setting, this means, the appearance prototypes being partitioned into $\boldsymbol{X}_1$ are more strongly associated with images that are partitioned in to $\boldsymbol{Y}_1$, and the relations between $\boldsymbol{X}_2$ and $\boldsymbol{Y}_2$ are similarly defined. The following objective function defines a reasonable criterion for the partitioning

$$J_1 = AS(X_1, Y_1) + AS(X_2, Y_2) - AS(X_1, Y_2) - AS(X_2, Y_1) \quad (1)$$

where $AS(X_1, Y_2) = \sum_{k \in X_1, l \in Y_2} w_{kl}$ and other three terms are similarly defined.

Maximizing J_1 is equivalent to maximizing the first two terms and minimizing the last two terms. The meaning of the criterion can be easily understood. Maximizing $AS(\boldsymbol{X}_1, \boldsymbol{Y}_1)$ means that images partitioned into sub set $\boldsymbol{Y}_1$ are strongly associated with the appearance prototypes being partitioned into the sub set $\boldsymbol{X}_1$. Maximizing $AS(\boldsymbol{X}_2, \boldsymbol{Y}_2)$ has similar explanation. Minimizing $AS(\boldsymbol{X}_1, \boldsymbol{Y}_2)$ means that images partitioned into sub set $\boldsymbol{Y}_2$ are least associated with appearance prototypes being partitioned into the sub set $\boldsymbol{X}_1$. Minimizing $AS(\boldsymbol{X}_2, \boldsymbol{Y}_1)$ can be understood similarly.

Although (1) is a sensible criterion, it could produce unbalanced cut in the sense that the size of any of the sub sets, $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$ can be very small even empty, because it can be easily shown that the criterion of (1) is equivalent to MinCut in graph theory. Whether other criterions, Ratio Cut, Min Max Cut and Normalized Cut will suit our current application needs further study. We here introduce another objective function which will produced a more balanced partition. First let us define a weight for each of the vertices in $\boldsymbol{X}$

$$w_x(k) = \sum_{\forall l} w_{kl} \qquad (2)$$

We then define the following objective function

$$J_2 = J_1 - \lambda \left(\sum_{k \in X_1} w_x(k) - \sum_{k \in X_2} w_x(k) \right)^2 \qquad (3)$$

where λ is a non-negative weighting constant. The new objective function is based on J_1 and a new term. The physical meaning of this new terms is that, we want to partition the appearance prototypes in such a way that, the total number of patterns accumulated over the whole database, should split equally between the two groups of appearance prototypes. If the database is large, this condition makes reasonable statistical sense. We will demonstrate in Section 4 that optimizing (3) does ensure that images in each group have very similar visual appearances, and moreover, the number of images is reasonably well distributed into each group. This is an interesting result given that we only impose the constraint that the numbers of image patterns fall into either of the groups should be equal, and not that the numbers of images fall into both groups should be the same.

3.2 A solution based on the Hopfield model

In order to partition the graph in such a way that J_2 is maximized, we here present a solution based on the Hopfield neural computational model [6]. We assign a binary variable to each vertex and for convenience using the following notations: $x_k = +1$ if $x_k \in \boldsymbol{X_1}$, $x_k = -1$ if $x_k \in \boldsymbol{X_2}$, $y_l = +1$ if $y_l \in \boldsymbol{Y_1}$, , $y_l = -1$ if $y_l \in \boldsymbol{Y_2}$, for $\forall$ *k, l* We now re-write (3) in terms of x_k and y_l:

$$J_2 = \sum_{k=1}^{m} \sum_{l=1}^{n} x_k y_l w_{kl} - \lambda \sum_{k=1}^{m} \left(x_k w_x(k) \right)^2 \qquad (4)$$

then, J_2 in (4) can be optimized by a Hopfield neural model. However, one difficulty of (4) is that λ has to be determined *a priori* (this is in general a difficult problem and no systematic solutions are available). To avoid this difficulty, we decided to optimize each term in (4) in turn. In order to prevent the algorithm getting trapped in local minimal, we optimize the terms in a stochastic manner:

Image and Feature Co-clustering Algorithm (IFCCA)

Step 1 Initialize x_k and y_l, $\forall$ *k, l,* to random numbers within (-1, 1)

Step 2 Randomly pick a vertex *k* (with a probability of $1/|\boldsymbol{X}|$) from $\boldsymbol{X}$ do:

$H_k = \sum_{\forall l} y_l w_{kl}$; $x_k = +1$ if $H_k \geq 0$; $x_k = -1$ if $H_k < 0$

Step 3 Randomly pick a vertex *l* (with a probability of $1/|\boldsymbol{Y}|$) from $\boldsymbol{Y}$ do:

$H_l = \sum_{\forall k} x_k w_{kl}$; $y_l = +1$ if $H_l \geq 0$; $y_l = -1$ if $H_l < 0$

Step 4 Randomly pick a vertex *j* (with a probability of $1/|\boldsymbol{X}|$) from $\boldsymbol{X}$ do:

$H_j = -\sum_{\forall l} x_l w_x(l)$; $x_j = +1$ if $H_j \geq 0$; $x_j = -1$ if $H_j < 0$

Step 5 if converge, stop, else go to **Step 2**

The algorithm first assigns random numbers to the states of the vertices. It then picks a random vertex from $\boldsymbol{X}$ and updates its state in such a way that the first term in (4) is increased. It then picks a random vertex from $\boldsymbol{Y}$ and updates it in such as way that the first term in (4) is increased. The algorithm then picks another random vertex from $\boldsymbol{X}$, this time the state of the vertex is updated to decrease the second term in (4). This process is repeated until either a pre-set maximum number of iterations is reached or until further changes in the vertices' states do not changes the objective function's value.

4 Simulation Results

We have tested the algorithm using an image database contained 6400 color photos, a subset from the commercially available Corel Photo CDs. Each image was represented by a 384-component appearance indexing feature as described in section 2.1. We then bi-partitioned the data in a recursive manner, and in each partition, the full set of appearance prototypes was used at all levels. At the first level, one group had 3047 and the other had 2696 images. At the second level, the numbers of images in each group were 2176, 1528, 1354 and 1342. At the third level, the numbers of images in each group were 1090, 1086, 839, 689, 618, 763, 717 and 625. At the 4th level, the numbers in each groups were 509, 581, 607, 479, 353, 486, 316, 373, 335, 283, 347, 389, 318, 399, 254 and 371. At the 5th level, the numbers of images in each group were 260, 249, 303, 278, 297, 310, 225, 254, 179, 174, 261, 225, 138, 178, 132, 241, 237, 98, 92, 191, 150, 197, 222, 167, 142, 176, 197, 202, 142, 112, 252, and 119. Although the original 6400 images were

divided into 64 categories (100 images each) of various themes, it is difficult to give meaningful objective measures of the clustering performances because some images in different categories actually contain very similar contents whilst some images in the same category can be very different. The only meaningful evaluation of the clustering performance is via subjective judgement. A good clustering method should cluster (subjectively) similar images into the same cluster. We have seen that the numbers of images in each cluster are spread quite evenly, and if the visual appearance of the images in each cluster is homogenous, then we can say the algorithm does a good job. Figure 2 shows example clusters from the 5th level of partitioning[1]. It is seen that each cluster contains visually similar images and the algorithm did do a very good job in grouping visually alike images into the same cluster. We have shown the clusters to a number of subjects and we have received very positive feedback. We have also observed that the appearance prototypes partitioned into $\boldsymbol{X}_1$ have strong correlation with the visual theme of the images partitioned into $\boldsymbol{Y}_1$ (and $\boldsymbol{X}_2$ with $\boldsymbol{Y}_2$), both in terms of the colors in the chromatic appearance prototypes and also in terms of the texture characteristics in the achromatic appearance prototypes.

5 Concluding remarks

In this paper, we have presented a method to model images and their content description features simultaneously. We presented a stochastic algorithm to jointly cluster images and their description features. We have presented experimental results which demonstrated that the algorithm did a very good job in clustering image with similar visual appearances into the same cluster. Unlike previous approaches to image clustering, our method associates features with clusters and explicitly identifies which features are more important to which cluster. We believe our method would be useful for developing advanced methods for content-based image database retrieval, e.g., building semantic-based image retrieval systems.

References

[1] A. W. M. Smeulders et al, "Content-based image retrieval at the end of the early years", IEEE Trans PAMI, vol. 22, pp. 1349 - 1380, 2000

[2] Y. Weiss, "Segmentation using eigenvectors: a unifying view", ICCV 1999

[3] J. Shi and J. Malik, "Normalized cut and image segmentation", IEEE PAMI, vol 22, pp. 888 – 905, 2000

[4] I. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioing", ACM Knowledge Discovery Data Mining KDD 01, pp. 269 – 274

[5] C. Ding etal, "A Min-max cut algorithm for graph partitioning and data clustering", IEEE 1st Conference on Data Mining, 2001, pp. 107 –114

[6] J. Hertz, R. G. Palmer and A. Koch, Introduction to the Theory of Neural Computation. Perseus Publishing, 1991

[7] G. Qiu, "Indexing chromatic and achromatic patterns for content-based image retrieval", Pattern Recognition, vol. 35, pp. 1675 – 1686, August, 2002

[8] G. Qiu, "Appearance indexing", Proc. ICASSP 2003, vol. III, pp. 597 – 600, April 2003

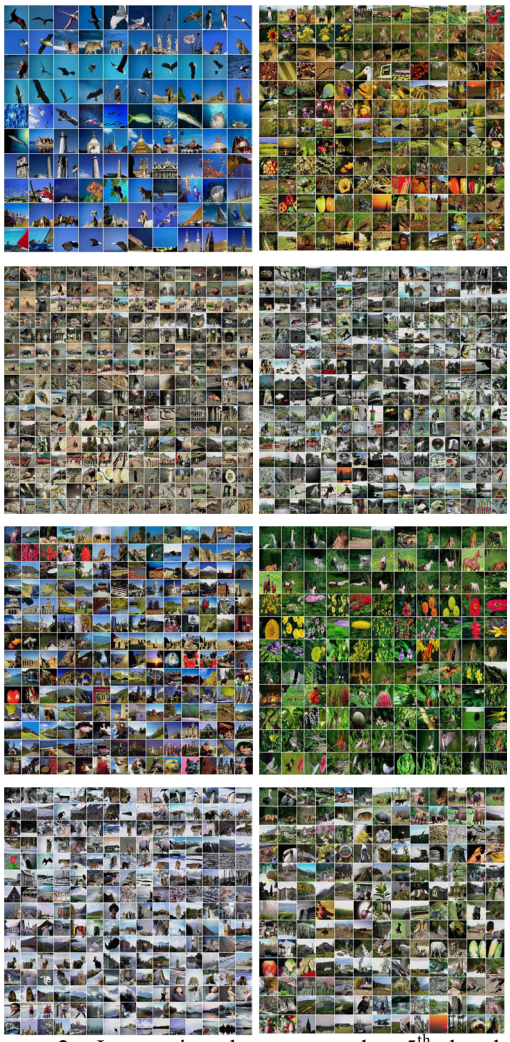

Figure 2: Image in clusters at the 5th level of partition. Images of all clusters will be made available on the web after the reviewing process.

[1] More results can be viewed on the author's website. http://www.cs.nott.ac.uk/~qiu/Research/NIPS2003/ifcc.html