Applying the Information Bottleneck Principle to Unsupervised Clustering of Discrete and Continuous Image Representations

Shiri Gordon
Faculty of Engineering
Tel-Aviv University, Israel
gordonha@post.tau.ac.il

Hayit Greenspan
Faculty of Engineering
Tel-Aviv University, Israel
hayit@eng.tau.ac.il

Jacob Goldberger
CUTe Systems Ltd.
Tel-Aviv, Israel
jacob@cute.co.il

Abstract

In this paper we present a method for unsupervised clustering of image databases. The method is based on a recently introduced information-theoretic principle, the information bottleneck (IB) principle. Image archives are clustered such that the mutual information between the clusters and the image content is maximally preserved. The IB principle is applied to both discrete and continuous image representations, using discrete image histograms and probabilistic continuous image modeling based on mixture of Gaussian densities, respectively. Experimental results demonstrate the performance of the proposed method for image clustering on a large image database. Several clustering algorithms derived from the IB principle are explored and compared.

1. Introduction

Image clustering and categorization is a means for high-level description of image content. The goal is to find a mapping of the archive images into classes (clusters) such that the set of classes provide essentially the same prediction, or information, about the image archive as the entire image set collection. The generated classes provide a concise summarization and visualization of the image content. Image archive clustering is important for efficient handling (search and retrieval) of large image databases [8, 3, 1]. In the retrieval process, the query image is initially compared with all the cluster centers. The subset of clusters that have the largest similarity to the query image is chosen, following which the query image is compared with all the images within this subset of clusters. Search efficiency is improved due to the fact that the query image is not compared exhaustively to all the images in the database.

Image clustering may be performed using discrete image representations (e.g. histograms) [8, 3] as well as continuous image representations (e.g. probabilistic continuous image modeling based on mixture of Gaussian densities) [7]. In recent work that compares between various image representation schemes, image modeling based on mixture of Gaussian densities was shown to outperform discrete image representations (such as the well-known color histograms, color correlograms, and more) [15]. In the current work we demonstrate unsupervised clustering in both the discrete and continuous image representations domains.

The clustering method presented in this work is based on the information bottleneck (IB) principle [14, 12, 10] (an earlier version was introduced in [6]). Characteristics of the proposed method include: 1) Image *models* are clustered rather than raw image pixels (image models may be discrete or continuous); 2) The IB method provides a simultaneous construction of both the clusters and the distance measure between them; 3) A natural termination of the bottom-up clustering process can be determined as part of the IB principle. This provides an automated means for finding the relevant number of clusters per archive; 4) The continuous agglomerative version of the IB clustering scheme is extended to include relaxation steps for better clustering results. The continuous probabilistic image modeling scheme is presented in section 2. The information bottleneck method along with clustering algorithms derived from the IB principle is presented in section 3. The method's application to discrete image representation is shown. In section 4 we extend the information bottleneck method to the case of continuous densities. Section 5 presents results of the proposed clustering method.

2. Grouping pixels into GMMs

In the first layer of the grouping process the raw pixel representation of an input image is shifted to a mid-level representation. The image representation may be discrete (e.g. histograms) or continuous. Histograms are well known in the literature and have been used substantially [13]. In

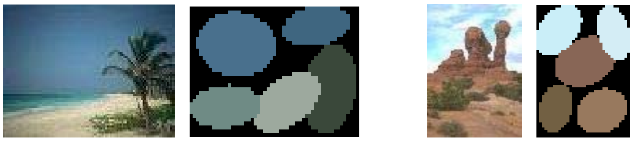

Figure 1. Input image (left). Image modeling via Gaussian mixture (right).

this section we briefly introduce the more recently proposed continuous image representation scheme.

In the continuous domain, each image is modeled as a mixture of Gaussians in the color ($L*a*b$) feature space. It should be noted that the representation model is a general one, and can incorporate any desired feature space (such as texture, shape, etc) or combination thereof. In order to include spatial information, the (x, y) position of the pixel is appended to the feature vector. Following the feature extraction stage, each pixel is represented with a five-dimensional feature vector, and the image as a whole is represented by a collection of feature vectors in the five-dimensional space. Pixels are grouped into homogeneous regions by grouping the feature vectors in the selected feature space. The underlying assumption is that the image colors and their spatial distribution in the image plane are generated by a mixture of Gaussians. Each homogeneous region in the image plane is thus represented by a Gaussian distribution, and the set of regions in the image is represented by a Gaussian mixture model.

The distribution of a d-dimensional random variable is a mixture of k Gaussians if its density function is:

$$f(y) = \sum_{j=1}^{k} \alpha_j \frac{1}{\sqrt{(2\pi)^d |\Sigma_j|}} \exp\{-\frac{1}{2}(y-\mu_j)^T \Sigma_j^{-1} (y-\mu_j)\}. \tag{1}$$

The Expectation-Maximization (EM) algorithm is used to determine the maximum likelihood parameters of a mixture of k Gaussians in the feature space (similar to [2]). The Minimum Description Length (MDL) principle [4] serves to select among values of k. In our experiments, k ranges from 4 to 8.

Figure 1 shows two examples of learning a GMM model for an input image. In this visualization each localized Gaussian mixture is shown as a set of ellipsoids. Each ellipsoid represents the support, mean color and spatial layout, of a particular Gaussian in the image plane.

3. The Information Bottleneck principle

The second layer of the image grouping process is based on information theoretic principle, the information bottleneck method (IB), recently introduced by Tishby *et al.* [14]. Using the IB method, the extracted image models are grouped, bottom-up, into coherent clusters. The IB principle states that among all the possible clusterings of the object set into a fixed number of clusters, the desired clustering is the one that minimizes the loss of mutual information between the objects and the features extracted from them. Assume there is joint distribution $p(x, y)$ on the "object" space X and the "feature" space Y. According to the IB principal we seek a clustering $\hat{X}$ such that, given a constraint on the clustering quality $I(X; \hat{X})$, the information loss $I(X; Y) - I(\hat{X}; Y)$ is minimized.

The IB principle can be motivated from Shannon's rate-distortion theory [4] which provides lower bounds on the number of classes we can divide a source given a distortion constraint. Given a random variable X and a distortion measure $d(x_1, x_2)$, defined on the alphabet of X, we want to represent the symbols of X with no more than R bits, i.e. there are no more than 2^R clusters. It is clear that we can reduce the number of clusters by enlarging the average quantization error. Shannon's rate-distortion theorem states that the minimum average distortion we can obtain by representing X with only R bits is given by the following distortion-rate function:

$$D(R) = \min_{p(\hat{x}|x) | I(X;\hat{X}) \leq R} Ed(x, \hat{x}) \tag{2}$$

where the average distortion $Ed(x, \hat{x})$ is $\sum_{x,\hat{x}} p(x)p(\hat{x}|x)d(x, \hat{x})$ and $I(X; \hat{X})$ is the mutual information between X and $\hat{X}$ given by:

$$I(X; \hat{X}) = \sum_{x,\hat{x}} p(x)p(\hat{x}|x) \log \frac{p(\hat{x}|x)}{p(\hat{x})}.$$

The random variable $\hat{X}$ can be viewed as a soft-probabilistic classification of X.

Unlike classical rate-distortion theory, the IB method avoids the arbitrary choice of a distance or a distortion measure. Instead, clustering of the object space X is done by preserving the relevant information about another space Y. We assume, as part of the IB approach, that $\hat{X} \rightarrow X \rightarrow Y$ is a Markov chain, i.e. given X the clustering $\hat{X}$ is independent of the feature space Y. Consider the following distortion function:

$$d(x, \hat{x}) = D_{KL}(\, p(y|X = x) \,||\, p(y|\hat{X} = \hat{x})\,) \tag{3}$$

where $D_{KL}(f||g) = E_f \log \frac{f}{g}$ is the Kullback-Liebler divergence [4]. Note that $p(y|\hat{x}) = \sum_x p(x|\hat{x})p(y|x)$ is a function of $p(\hat{x}|x)$. Hence, $d(x, \hat{x})$ is not predetermined. Instead it depends on the clustering. Therefore, as we search for the best clustering we also search for the most suitable distance measure.

The loss in the mutual information between X and Y caused by the (probabilistic) clustering $\hat{X}$ can be viewed as the average of this distortion measure:

$$I(X;Y) - I(\hat{X};Y) = \sum_{x,\hat{x},y} p(x,\hat{x},y) \log \frac{p(x|y)}{p(x)}$$

$$- \sum_{x,\hat{x},y} p(x,\hat{x},y) \log \frac{p(y|\hat{x})}{p(y)} = \sum_{x,\hat{x},y} p(x,\hat{x},y) \log \frac{p(y|x)}{p(y|\hat{x})}$$

$$= \sum_{x,\hat{x}} p(x,\hat{x}) \sum_{y} p(y|x) \log \frac{p(y|x)}{p(y|\hat{x})}$$

$$= ED_{KL}(p(y|x)||p(y|\hat{x})).$$

Substituting distortion measure (3) in the distortion rate function (2) we obtain:

$$D(R) = \min_{p(\hat{x}|x)|I(X;\hat{X}) \leq R} I(X;Y) - I(\hat{X};Y) \qquad (4)$$

which is exactly the minimization criterion proposed by IB principle, namely, finding a clustering that causes minimum reduction of the mutual information between the objects and the features.

3.1. Clustering algorithms based on the IB principle

The minimization problem posed by the IB principle can be approximated by a greedy algorithm based on a bottom-up merging procedure [12]. The algorithm starts with the trivial clustering where each cluster consists of a single point. In order to minimize the over all information loss caused by the clustering, classes are merged in every (greedy) step such that the loss in the mutual information caused by merging them is the smallest. Let c_1 and c_2 be two clusters of symbols from the alphabet of X, the information loss due to the merging of c_1 and c_2 is:

$$d(c_1, c_2) = I(C_{before}, Y) - I(C_{after}, Y) \geq 0$$

where $I(C_{before}, Y)$ and $I(C_{after}, Y)$ are the mutual information between the classes and the feature space before and after c_1 and c_2 are merged into a single class. Standard information theory manipulation reveals:

$$d(c_1, c_2) = \sum_{y,i=1,2} p(c_i, y) \log \frac{p(c_i, y)}{p(c_i)p(y)}$$

$$- \sum_{y} p(c_1 \cup c_2, y) \log \frac{p(c_1 \cup c_2, y)}{p(c_1 \cup c_2)p(y)}$$

$$= \sum_{y,i=1,2} p(c_i, y) \log \frac{p(y|c_i)}{p(y|c_1 \cup c_2)}$$

$$= \sum_{y,i=1,2} p(c_i) D_{KL}(p(y|c_i)||p(y|c_1 \cup c_2)). \qquad (5)$$

Hence, the distance measure between clusters c_1 and c_2, derived from the IB principle, takes into account both the dissimilarity between the distribution $p(y|c_1)$ and $p(y|c_2)$ and the size of the two clusters.

The greedy AIB algorithm arranges the objects in a tree structure, which has many advantages for database management. The algorithm also enables to define the optimal number of clusters that represent the objects in the database. However, the main obstacle to the greedy agglomerative procedure is that finding an optimal clustering solution is not guaranteed. In fact, it is not guaranteed to find a stable solution, in which each object belongs to the cluster it is most similar to. The issue of cluster optimization is common in many (both top-down and bottom-up) hierarchical clustering techniques. These techniques, due to their greedy nature, often require additional relaxation steps for cluster optimization [8].

An augmented AIB algorithm is proposed that combines the AIB with algorithms that perform cluster optimization in each of the tree levels. The sequential IB (SIB) clustering algorithm [10] and the K-means algorithm [5] are two algorithms that can be used for this purpose. The SIB is a modification of the standard K-means algorithm. Like the K-means procedure, the sequential clustering maintains a fixed amount of K clusters. The algorithm starts from an initial partition C of the objects in X into clusters. At each step of the algorithm one object $x \in X$, is "drawn" out of its current cluster $c(x)$ into a new singleton cluster. Using a greedy agglomerative step, x is merged into c^{new} so that $c^{new} = argmin_{c \in C} d(\{x\}, c)$ and a new partition C^{new} is obtained. The main difference between sequential clustering and the standard K-means is in the updating scheme. The K-means algorithm performs *parallel updates*: Only after each element x selects its new cluster, do we move *all* the elements to their new clusters. The cluster centers are therefore updated *once*, after all the elements were moved to their preferred location. In the sequential clustering algorithm the cluster centers are modified after *each* element selects its new cluster. The sequential updating scheme accelerates the convergence process.

When applying the sequential algorithm and the K-means algorithm to the IB method, the greedy merging criterion, $d(\{x\}, c)$, is the information loss due to the merging of two clusters (Equation (5)). The score function being maximized in each of the algorithm iterations is the mutual information, $I(C;Y)$. Since $I(C;Y)$ is known to be upper bounded [4], convergence to a local maximum is guaranteed.

3.2. Applying the IB principle to discrete image representation

The IB principle has been used for clustering in a variety of discrete domains, including documents [12, 10], galaxies [11] and neural codes [9]. In this work we apply the IB principle for clustering in the image domain. We start by applying the IB principle to clustering of discrete image representations. In particular, we use global color histograms for both image and cluster representation.

In the following we denote by X the set of images we want to classify. We assume a uniform prior probability $p(x)$ of observing an image. Denote by Y the random variable associated with the feature vector extracted from a single pixel. The image histogram is then used to describe the feature distribution within an image, $f(y|x)$. The next step is to define the distribution of the features within a cluster of images: $f(y|c)$. This is done by simply averaging the histograms of the individual images within the cluster.

Let $p_1 = \{p_{11}, p_{12}, \ldots, p_{1m}\}$,$p_2 = \{p_{21}, p_{22}, \ldots, p_{2m}\}$ be the histograms associated with image clusters c_1, c_2 respectively. The histogram of the merged cluster $c_1 \cup c_2$ is:

$$p = \frac{|c_1|}{|c_1 \cup c_2|} p_1 + \frac{|c_2|}{|c_1 \cup c_2|} p_2.$$

According to expression (5), the distance between the two image clusters c_1 and c_2 is:

$$d(c_1, c_2) = \sum_{i=1,2} \frac{|c_i|}{|X|} D_{KL}(p_i || p) \tag{6}$$

where $|X|$ is the size of the image database. The discrete KL distance $D_{KL}(p_1||p)$ is computed using the following equation:

$$D_{KL}(p_1||p) = \sum_{j=1}^{m} p_{1j} \log \frac{p_{1j}}{p_j}, \tag{7}$$

$D_{KL}(p_2||p)$ is computed in a similar manner.

4. Applying the IB principle to continuous distributions

In this section we generalize the IB principle to the case where a continuous feature set is endowed with a mixture of Gaussians distribution. Given the image set X and the feature set Y, the Gaussian mixture model we use to describe the feature distribution within an image, x, is exactly the conditional density function $f(y|x)$. Assuming a uniform prior probability $p(x)$ of observing an image, we have a joint image-feature distribution $p(x, y)$. Let c be a cluster of images where each image $x \in c$ is modeled via a GMM:

$$f(y|x) = \sum_{j=1}^{k(x)} \alpha_{x,j} N(\mu_{x,j}, \Sigma_{x,j}) \qquad x \in c$$

such that $k(x)$ is the number of Gaussian components in $f(y|x)$. The distribution of the features within a cluster of images $f(y|c)$, is obtained by averaging all the image models within the cluster:

$$f(y|c) = \frac{1}{|c|} \sum_{x \in c} f(y|x) = \frac{1}{|c|} \sum_{x \in c} \sum_{j=1}^{k(x)} \alpha_{x,j} N(\mu_{x,j}, \Sigma_{x,j}). \tag{8}$$

Note that since $f(y|x)$ is a GMM distribution, the density function per cluster c, $f(y|c)$, is a mixture of GMMs and therefore it is also a GMM.

Let $f(y|c_1), f(y|c_2)$ be the GMMs associated with image clusters c_1, c_2 respectively. The GMM of the merged cluster $c_1 \cup c_2$ is:

$$f(y|c_1 \cup c_2) = \frac{1}{|c_1 \cup c_2|} \sum_{x \in c_1 \cup c_2} f(y|x)$$

$$= \sum_{i=1,2} \frac{|c_i|}{|c_1 \cup c_2|} f(y|c_i).$$

The distance between the two image clusters c_1 and c_2, as derived from expression (5), is:

$$d(c_1, c_2) = \sum_{i=1,2} \frac{|c_i|}{|X|} D_{KL}(f(y|c_i) || f(y|c_1 \cup c_2)) \tag{9}$$

where $|X|$ is the size of the image database. Hence, to compute the distance between two image clusters, c_1 and c_2, we need to compute the KL distance between two GMM distributions.

Since the KL distance between two GMMs can not be analytically computed, we can numerically approximate it through Monte-Carlo procedures. Denote the feature set extracted from the images that belongs to cluster c_1 by $y_1 \ldots y_n$. The KL distance $D_{KL}(f(y|c_1)||f(y|c_1 \cup c_2))$ can be approximated as follows:

$$D_{KL}(f(y|c_1)||f(y|c_1 \cup c_2)) \cong \frac{1}{n} \sum_{t=1}^{n} \log \frac{f(y_t|c_1)}{f(y_t|c_1 \cup c_2)}. \tag{10}$$

Another possible approximation is to use synthetic samples produced from the Gaussian mixture distribution $f(y|c_1)$ instead of the image data. This enables us to compute the KL distance without referring to the images from which the models were built. Image categorization experiments show no significant difference between these two proposed approximations of the KL distance [7]. The expression

$D_{KL}(f(y|c_2)||f(y|c_1 \cup c_2))$ can be approximated in a similar manner.

The agglomerative IB algorithm for image clustering is the following:

1. Start with the trivial clustering where each image is a cluster.
2. In each step merge clusters c_1 and c_2 such that information loss $d(c_1, c_2)$ (Equation 9) is minimal.
3. Continue the merging process until the information loss $d(c_1, c_2)$ is more than a predefined threshold, indicating that we attempt to merge two non-similar clusters.

The AIB algorithm may be augmented by utilizing cluster optimization algorithms, such as the SIB and the K-means algorithms (section 3). Implementation of the augmented AIB procedure on a database of image GMMs, requires the following considerations: First, in the high levels of the tree created by the AIB algorithm, the number of clusters K is small. Many images are thus affiliated with each cluster. As a result the clusters models (centroids) are generated from a very large set of GMMs, and become very complex and fuzzy (Equation 8)[1]. Due to the centroids fuzziness, when the optimal classification is reached, images can be close to more than one centroid. It is thus very difficult to reach a definite classification result, in which there are no more changes in cluster grouping.

Second, the Monte-Carlo procedure used for estimating the KL-distance requires a very large sample set for representing all the Gaussians in the cluster centroid. This leads to a very high computational complexity, and makes the calculation sensitive to sample noise. When an image is close to more than one cluster centroid, using the estimated KL-distance in the greedy classification criterion can make the image shift from one cluster to the other in the algorithm iterations. In such a case the entire classification result is unstable around the optimal point.[2]

In order to address the above-described limitations, an image is transitioned from one cluster to the other only if the difference between the image and the new cluster centroid is smaller than the distance between the image and its current cluster centroid by a predefined threshold. A change is thus performed only if it causes a significant reduction in information loss.

A stopping criteria for the algorithm iterations is required. We use the mutual information $I(C;Y)$, between the image clustering C and the feature set Y, as the stopping criteria. Trying to maximize the mutual information created by the various partitions (in each of the algorithm iterations), we let the algorithm iterate as long as the mutual information increases. Using mutual information as a stopping criterion is not straight-forward when the features are endowed with a mixture of Gaussian distribution. No closed-form expression exists in that case. The successive merging process performed in the AIB algorithm can give us, as a byproduct, an approximation method for $I(C;Y)$. Cluster models are merged successively into a single cluster, merging two clusters at a time according to Equation 9 (The merging order is of no importance). The information loss calculated in each step is accumulated. The *total* information loss during the merging process is exactly the mutual information $I(C;Y)$ we wish to approximate.

5. Results

This section presents an investigative analysis of the IB method for image clustering. Experimental results demonstrate the IB method's ability to discover an optimal number of clusters in the database using the AIB algorithm. Retrieval experiments are used to evaluate the clustering quality of the proposed method and of the various clustering algorithms introduced. The database used throughout the experiments consists of 1460 images selectively hand-picked from the COREL database to create 16 categories. The images within each category have similar colors and color spatial layout, and can be labelled with a high-level semantic description.

The AIB clustering method described in section 4, was applied to our database of 1460 images. The clustering is performed on the GMM image representation. We started with 1460 clusters where each image model is a cluster. After 1459 steps all the images were grouped into a single cluster. The given database was thus arranged in a tree structure. The loss of mutual information during each merging step of the clustering process is shown in Figure 2. The x-axis indicates the last 60 steps of the algorithm. The y-axis shows the amount of mutual information loss (in bits) caused by merging the two clusters selected at the corresponding step. The labels associated with the last points of the graph indicate the number of clusters created in the corresponding step. There is no need to present the information loss during the entire clustering process, since meaningful changes occur only towards the end of the process. There is a gradual increase in information loss until we reach a point of significant loss of information. This point helps us determine a "meaningful" number of clusters existing in the database. From this point on, every merge causes a significant degradation of information and therefore leads to a worse clustering scenario. As can be seen from Figure 2, the

[1] The model requires the parameters of $\sum_{x \in c} k(x)$ Gaussian distributions.

[2] Note that this problem doesn't exist when calculating the KL-distance between discrete distributions. The discrete KL-distance has a closed form solution (Equation 7) and the stochastic process associated with the Monte-Carlo procedure is not required.

first significant jump in the graph is found in the transition from 13 to 12 clusters.

Figure 3 presents 5 sample images from each of the 13 "meaningful" clusters created by the AIB algorithm. The GMM generated for each cluster is shown on the right [3]. A Gaussian in the model is displayed as a localized colored ellipsoid. Some of the Gaussians overlap spatially and thus are not explicitly shown in the image. A clear distinction between the groups is evident in the Gaussian mixture characteristics, in blob color features and their spatial layouts. Progressing an additional step of the algorithm, towards 12 clusters, results in the merging of clusters C_{12} and C_{13}. We note that the two clusters appear rather different. The visual inhomogeneity is consistent with the significant loss of information, as indicated via the information loss criterion.

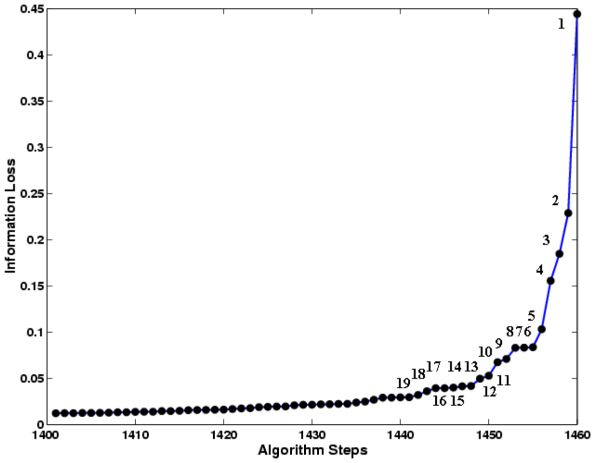

Figure 2. Loss of mutual information during the AIB clustering process. The labels attached to the final 19 algorithm steps, indicate the number of clusters formed per step.

In the following experiments we use image retrieval to evaluate clustering quality. We first evaluate the proposed IB clustering methodology by comparing it to another clustering methodology based on Histogram Intersection. We then evaluate the clustering quality of the various cluster optimization algorithms introduced in section 4. During the retrieval process the query image is first compared with all cluster models. The clusters most similar to the query image are chosen. The query image is next compared with all the images within these clusters.

Retrieval results are evaluated by precision versus recall (PR) curves. Recall measures the ability of retrieving all relevant or perceptually similar items in the database. It

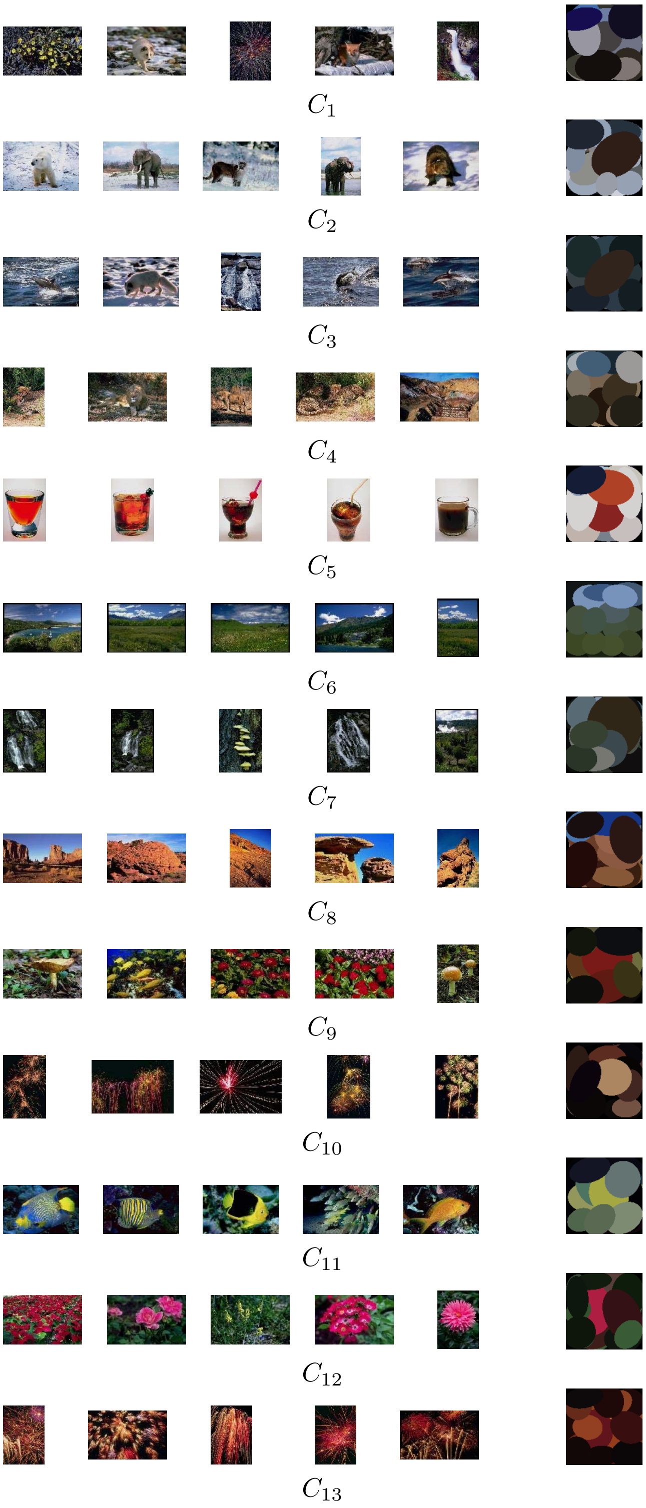

Figure 3. Example images for the 13 clusters created by the AIB algorithm from the 1460 image database. A GMM model generated for each cluster is shown on the right.

[3] A color version may be found in http://www.eng.tau.ac.il/ ~ hayit.

is defined as the ratio between the number of perceptually similar items retrieved and the total relevant items in the database. Precision measures the retrieval accuracy and is defined as the ratio between the number of relevant or perceptually similar items retrieved and the total number of items retrieved. A better PR curve indicates a better clustering since the query is compared only with the images within the closest clusters. The more these clusters are correlated with the labelled categories, the better the PR curve will be.

Retrieval results for 320 images were averaged in all of the experiments, 20 images drawn randomly from each of the 16 labelled categories we have in the database. PR curves were calculated for 10,20,30,40,50, and 60 retrieved images. The database was divided into 13 clusters in all of the experiments. Note that this is a single point in the agglomerative clustering process.

We first wish to evaluate the performance of the IB clustering methodology. Global color histograms are used to represent the images. The agglomerative IB clustering is compared to the agglomerative clustering based on Histogram Intersection (H.I.) (similar to [8]). Thirteen clusters are generated by each clustering methodology. The distance measure used in the retrieval process is the discrete KL distance in the AIB clustering case, and the H.I. distance in the agglomerative H.I. clustering. Retrieval based on initial clustering is compared to exhaustive search in both cases. A comparison is conducted using the following PR curves: PR for retrieval based on clustering using the IB method and the KL-distance (solid line), PR for exhaustive retrieval using KL-distance (dashed line), PR for retrieval based on clustering using H.I., both for clustering and for retrieval (dash-dot line) and exhaustive retrieval using H.I. (dotted line).

The four PR curves can be seen to split into two groups. The top two curves correspond to retrieval using the discrete KL distance measure and the bottom two curves correspond to retrieval using the H.I. distance measure. Within each group one curve presents the results of retrieval in a clustered dataset and the second curve presents the exhaustive retrieval results. A comparison between the two retrieval distance measures is enabled by looking at the two PR curves of the exhaustive search. Such a comparison indicates that the information-theoretic KL distance achieves better results than the H.I. measure. Investigating the clustering methodologies is enabled by a comparison of the two PR curves that reflect retrieval in a clustered dataset. Clustering based on the AIB algorithm provides the best retrieval results. Retrieval using clustering based on H.I. achieved poor performance. It is interesting to note that these results are even worse than the related exhaustive retrieval case (using the H.I. as a distance measure). These results indicate a strong advantage for using the information-theoretic tools of AIB for clustering and KL distance for retrieval.

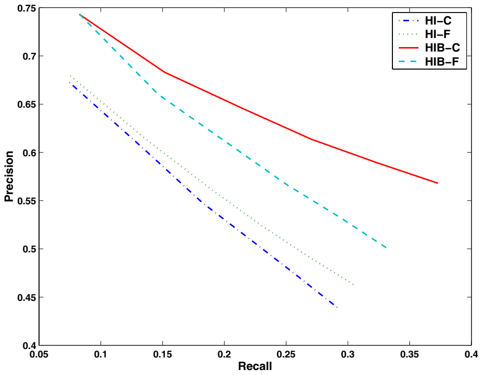

Figure 4. Precision vs recall for evaluating the IB method relative to a method based on H.I. Cluster search based on H.I. (dash-dot line, HI-C). Exhaustive search based on H.I. (dotted line, HI-F). Cluster search based on IB method and KL-distance (solid line, HIB-C). Exhaustive search based on KL-distance (dashed line, HIB-F). All experiments use the color histogram representation.

In a second experiment we wish to examine the performance of the AIB algorithm and compare its performance to the augmented AIB algorithms, using SIB and K-means. The various unsupervised clustering algorithms are tested in the continuous domain, using the GMM cluster representation. We initiate both the SIB and the K-means with the results obtained by the AIB for 13 clusters. Since the AIB is a greedy algorithm we expect that the iterations performed by the SIB and the K-means algorithms will improve the clustering results. Figure 5 summarizes the comparison results. The SIB results are plotted as a dashed line. The K-means results are plotted as a solid line and the AIB results are plotted as a dotted line. The KL-distance is used as the distance measure during the retrieval process. The performance of the SIB and the K-means algorithms are better than those of the AIB algorithm, thus encouraging post-processing of the AIB clustering results via cluster optimization.

6. Discussion

We have presented the IB method for unsupervised clustering of image databases. The unsupervised clustering scheme is based on information theoretic principles. It provides image-sets for a concise summarization and visualization of the image content within a given image archive. Applying the IB principle for clustering of image archives, using either discrete or continuous image representations,

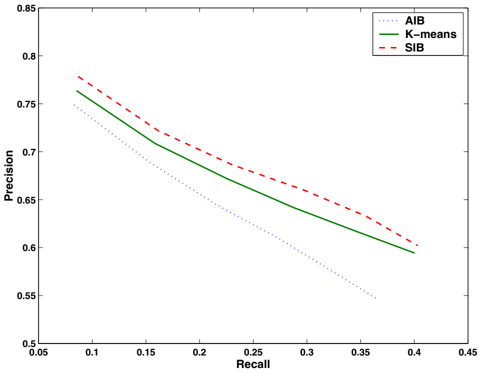

Figure 5. Precision vs recall for evaluating clustering results created with different algorithms. Clustering by AIB algorithm (dotted line). Clustering by K-means algorithm (solid line). Clustering by SIB algorithm (dashed line). All experiments use the GMM image representation.

is novel. So is the ability of the method to define a "meaningful" number of clusters that exist in the database (this number is an important parameter for many clustering algorithms).

Retrieval results indicate a strong advantage for using the information-theoretic tools of AIB for clustering and KL distance for retrieval. It was demonstrated that the greedy AIB algorithm results can be improved by using cluster optimization, via relaxation algorithms, such as the SIB and K-means.

There are several issues related to our framework that still need to be addressed. Using the Monte-Carlo procedure for the KL-distance approximation is problematic for the case of a GMM with a large number of Gaussians. In that case a large number of samples is required for the approximation, increasing the complexity and the probability for sampling noise. Further effort should be dedicated to finding a more compact cluster representation (i.e. a GMM with a reduced number of parameters). An analytical solution, or a simpler estimation, for the calculation of the KL-distance between two GMMs is also desirable.

Image variations including illumination irregularities, texture and other artifacts are not accounted for in the models used. The additional features influence on clustering quality should be investigated. Future work entails making the current method more feasible for large databases and using the tree structure created by the AIB algorithm, for the creation of a "user friendly" browsing environment.

References

[1] K. Barnard, P. Duygulu, and D. Forsyth. Clustering art. In *Computer Vision and Pattern Recognition (CVPR 2001)*, Hawaii, December 2001.

[2] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[3] J. Chen, C.A. Bouman, and J.C. Dalton. Hierarchical browsing and search of large image databases. *IEEE transactions on Image Processing*, 9(3):442–455, March 2000.

[4] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, New York, 1991.

[5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley, New York, 2001.

[6] J. Goldberger, H. Greenspan, and S. Gordon. Unsupervised image clustering using the information bottleneck method. In *DAGM*, 2002.

[7] H. Greenspan, J. Goldberger, and L. Ridel. A continuous probabilistic framework for image matching. *Journal of Computer Vision and Image Understanding*, 84:384–406, 2001.

[8] S. Krishnamachari and M. Abdel-Mottaleb. Hierarchical clustering algorithm for fast image retrieval. In *SPIE Conference on Storage and Retrieval for Image and Video databases VII*, pages 427–435, San-Jose, CA, Jan 1999.

[9] E. Schneidman, N. Slonim, N. Tishby, R. R. deRuyter van Steveninck, and W. Bialek. Analysing neural codes using the information bottleneck method. In *Advances in Neural Information Processing Systems, NIPS*, 2001.

[10] N. Slonim, N. Friedman, and N. Tishby. Unsupervised document classification using sequential information maximization. In *Proc. of the 25-th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002.

[11] N. Slonim, R. Somerville, N. Tishby, and O. Lahav. Objective classification of galaxy spectra using the information bottleneck method. 323:270–284, 2001.

[12] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *Proc. of Neural Information Processing Systems*, pages 617–623, 1999.

[13] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[14] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[15] N. Vasconcelos and A. Lippman. Feature representations for image retrieval: beyond the color histogram. In *Proc. of the Int. Conference on Multimedia and Expo*, New York, August 2000.