Concepts Learning with Fuzzy Clustering and Relevance Feedback

Bir Bhanu* and Anlei Dong

Center for Research in Intelligent Systems, University of California, Riverside, California 92521, USA

Tel: +1-909-787-3954; fax: +1-909-787-3188

Abstract

In recent years feedback approaches have been used in relating low-level image features with concepts to overcome the subjective nature of the human image interpretation. Generally, in these systems when the user starts with a new query, the entire prior experience of the system is lost. In this paper, we address the problem of incorporating prior experience of the retrieval system to improve the performance on future queries. We propose a semi-supervised fuzzy clustering method to learn class distribution (meta knowledge) in the sense of high-level concepts from retrieval experience. Using fuzzy rules, we incorporate the meta knowledge into a probabilistic feature relevance feedback approach to improve the retrieval performance. Results on synthetic and real databases show that our approach provides better retrieval precision compared to the case when no retrieval experience is used.

Keywords: Content-based retrieval; Image databases; Multiple concepts; Learning; Indexing; Meta knowledge

1. Introduction

Past several years have witnessed the developments of a variety of content-based retrieval methods and systems for image databases. In interactive relevance learning approaches (Peng *et al.*, 1999; Rui *et al.*, 1998; Minka and Picard, 1997) for image databases, a retrieval system dynamically adapts and updates the relevance of the images to be retrieved. In these systems, images are generally represented by numeric features or attributes, such as texture, color and shape, which are called low-level visual features (Flickner *et al.*, 1995). What user desires are called human high-level concepts. The task of relevance feedback learning is to reduce the gap between low-level visual features and high-level human concepts.

The most important thing to be learned in relevance feedback learning are the weights of different features. Learning a user's ideal query is also important. The feedback, provided by different users in the form of “similar” (positive) images and “dissimilar” (negative) images, is an important part of the experience. In these systems,

generally once the user is done with a query and starts a new query, the experience (meta knowledge) gained by the systems with previous queries is lost. For this scenario, there is only user adaptation but no long-term learning. It is possible to exploit the system's experience for learning visual concepts. Meta knowledge is the experience of each query image with various users. This experience consists of the classification of each image into various classes (clusters), relevances (weights) of features and the number of times this image is selected as a query and marked as positive or negative.

In practical applications, we desire good retrieval performance not for a single user, but for many users. Here, good retrieval performance means high precision and fast response. Although different people may associate the same image into different categories, the generalization of viewpoints of many people count much for making this decision and it will help in indexing large image databases. This paper attempts to capture and utilize the previous experiences of the system with various queries to learn visual concepts. The visual concepts are continually learned and refined over time, not necessarily from the interaction with one single user in a single retrieval session.

At the very beginning, images in the database have no high-level conceptual information. With more and more users performing retrieval tasks, based on their feedback, it is possible for the system to capture this experience and learn image class distribution in the sense of high-level concepts obtained during the earlier experience of image retrieval. This method can give better results than those which are purely based on low-level features since we have extra knowledge of high-level classification. This can significantly improve system performance which includes both the instantaneous performance and the performance at each iteration of relevance feedback.

The above discussion raises two fundamental questions: (A). How to learn class distribution in the sense of high-level concepts from different users' queries and associated retrievals? (B). How to develop a better relevance learning method by integrating low-level features and high-level class distribution knowledge?

The key contribution of the paper is to present a new approach to address both of these questions. Based on the semi-supervised fuzzy c-means (SSFCM) clustering (Petrycz and Waletzky, 1997), we propose a modified fuzzy clustering method which can effectively learn class distribution (meta knowledge) in the sense of high-level concept from retrieval experience. Using fuzzy rules, we incorporate the meta knowledge into a probabilistic relevance feedback method to improve the retrieval performance. As stated above, meta knowledge consists of a variety of knowledge extracted from prior experience of the system.

This paper is organized as follows. Section 2 describes the related research on learning visual concepts. Section 3 gives our technical approach (Algorithm A) for improving retrieval performance by incorporating meta knowledge into relevance feedback method. Here the assumption is that retrieval experience is directly given in matrix form. Section 4 presents the improved approach (Algorithm B) which derives retrieval experience by using a probabilistic technique and modifies concept learning and relevance feedback (Bhanu and Dong, 2001). Experimental results of these two algorithms are provided in Section 5 and Section 6 presents the conclusions of the paper.

2. Related work

Since there is a big gap between high-level concepts and low-level image features, it is difficult to extract semantic concepts from low-level features. Chang *et al.* (1998) propose the idea of semantic visual templates (*SVT*), where templates represent a personalized view of a concept. The system interacting with the user generates a set of queries to represent the concept. However, the system does not accommodate multiple concepts which may be present in a single image and their interactions. Tieu and Viola (1999) use a *boosting* technique to learn a classification function when a user selects a few example images at query time. The classifier relies on 20 of the large number of visual features. Cox *et al.* (2000) use a Bayesian approach for optimal solution for multiple visual features. Ratan *et al.* (1999) adopt multiple instance-learning paradigm using the diverse density algorithm to model the ambiguity in images and to learn visual concepts. This method requires image segmentation, which leads to additional preprocessing and the brittleness of the method. Rui *et al.* (2000) optimize learning process using a hierarchical feature model. This approach yields explicit optimal solutions and it is fast to compute. All the above mentioned systems attempt to learn human concepts only with a single user.

Lipson et al. (1997) use qualitative spatial and photometric relationships to encode class models for classifying scenes by adopting a *configural recognition* scheme. Lim (1999) proposes the notion of visual keywords for content-based retrieval, which can be adapted to visual content domain via learning from examples. The examples are generated by a human during off-line. The keywords of a given visual content domain are visual entities used by the system. In this both of these two approaches, no relevance feedback is used.

Unlike the previous research published to date, this paper exploits meta knowledge, accumulated over time incorporating experience of many users on various queries, for learning visual concepts where multiple concepts may be present in the same query image. Fuzzy clustering and relevance feedback are the main tools used for this purpose.

3. Technical approach

Fig. 1 illustrates our approach for concept learning by exploiting meta knowledge. Since it is not uncommon that one image can be ascribed into different concepts, we use semi-supervised fuzzy c-means clustering method to learn the concept distribution, and the images' ascriptions to different concepts are represented by the resulting partition matrix. Initially, when the system is presented with a query image, it does not know which concept the user is seeking. It just presents the images to the user using the *K*-NN search on the entire database. If the user is not satisfied with these retrievals and provides feedback, the system attempts to decide the concept that is sought by the user.

The concept distribution knowledge is derived from semi-supervised fuzzy clustering performed over time. If the desired concept is achieved, the system only needs to search images within the cluster corresponding to this concept; otherwise, it performs statistical relevance learning to estimate feature weights and search images in the entire database. With increased retrieval experiences, the concept learning is improved, which helps to capture user's desired concept more precisely, and thus future retrieval performance is improved. Fig. 2 provides the detailed system block diagram. The focus is the upper-right (dotted) region. The rest of the components shown in the figure represent a typical probabilistic feature relevance learning system.

In this section, we present a concept learning algorithm (we call it Algorithm A) based on the assumption that the retrieval experience is directly represented by positive matrix and negative matrix, which we will introduce in Section 3.1.

3.1. Problem formulation

Assume each image corresponds to a pattern in the feature space $\mathbf{R}^n$. The set of all the patterns is $\boldsymbol{X}$. We also assume the number of high-level classes c is known. After the image database (size N) has already experienced some retrievals by different users, we have $\boldsymbol{X} = \boldsymbol{X}^u \cup \boldsymbol{X}^p \cup \boldsymbol{X}^n$, where $\boldsymbol{X}^u$ represents the set of the images that are never marked (unmarked) by users in the previous retrievals; $\boldsymbol{X}^p$ represents the set of the images that are marked positive by users; $\boldsymbol{X}^n$ represents the set of the images that are marked negative by users. Note: $\boldsymbol{X}^p \cap \boldsymbol{X}^n \neq \varnothing$. The reason is that one image may be marked positive in one retrieval while marked negative in another. Even though two or more retrievals may actually be for the same high-level concept (cluster), it is still possible that the image is marked both

positive and negative since whether or not to associate an image to a specific high-level concept is subjective to different users. We provide two matrices to represent the previous retrieval experience:

(i) *positive matrix* $P = [\, p_{ik} \,]_{c\times N}$: if image k is ever marked positive for the ith cluster n^+ times, the element $p_{ik} = n^+$; otherwise, $p_{ik} = 0$;

(ii) *negative matrix* $Q = [\, q_{ik} \,]_{c\times N}$: if image k is ever marked negative for the ith cluster n^- times, the element $q_{ik} = n^-$; otherwise, $q_{ik} = 0$.

Our problem is how to use the retrieval experience to improve the fuzzy clustering performance, i.e., make the data partition closer to a human's high-level concept.

3.2. Fuzzy clustering

The fuzzy clustering method (Jain *et al.*, 1999; Bezdek *et al*., 1999; Gustafson and Kessel, 1978) is a data analysis tool concerned with the structure of the dataset under consideration. The clustering result is represented by grades of membership of every pattern to the classes established. Unlike binary evaluation of crispy clustering, the membership grades in fuzzy clustering are evaluated within the [0, 1] interval. The necessity of fuzzy clustering lies in the reality that a pattern could be assigned to different classes (categories). The objective function method is one of the major techniques in fuzzy clustering. It usually takes the form

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{a} \left\| x_k - v_i \right\|^2 \tag{1}$$

where x_k, $k = 1, 2, \ldots, N$, are the patterns in $\mathbf{R}^n$, $v_1, v_2, \ldots, v_c$ are prototypes of the clusters, $1 < a < \infty$, and $U = [u_{ik}]$ is a partition matrix describing clustering results whose elements satisfy two conditions:

(a) $\sum_{i=1}^{c} u_{ik} = 1$, $k = 1, 2, \ldots, N$;

(b) $u_{ik} \geq 0$, $i = 1, 2, \ldots, c$ and $k = 1, 2, \ldots, N$.

The task is to minimize J with respect to the partition matrix and the prototypes of the clusters, namely $\min_{v_1, v_2, \cdots v_c, U} J$, with U satisfying conditions (a) and (b). The distance function in (1) is the Mahalanobis distance defined as

$$d_{ik}^2 = \|x_k - v_i\|^2 = (x_k - v_i)^T W (x_k - v_i) \tag{2}$$

where W is a symmetrical positive definite matrix in $\mathbf{R}^n \times \mathbf{R}^n$.

The fuzzy c-means (FCM) method is often frustrated by the fact that lower values of J do not necessarily lead to better partitions. This actually reflects the gap between numeric-oriented feature data and classes understood by humans. The semi-supervised FCM method attempts to overcome this limitation (Pedrycz and Waletzky, 1997; Bensaid *et al.*, 1996) when the labels of some of the data are already known.

3.2.1. Semi-supervised c-means fuzzy clustering

Pedrycz and Waletzky (1997) modified objective function J given by (1) as

$$J_1 = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^2 d_{ik}^2 + \alpha \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ik} - f_{ik} b_k)^2 d_{ik}^2 \tag{3}$$

where b_k =1 if x_k is labeled, and b_k = 0 otherwise, k = 1, 2, …, N. The matrix $F = [f_{ik}]_{cxN}$ with the given label vectors in appropriate columns and zero vectors elsewhere. α ($\alpha \geq 0$) denotes a scaling factor whose role is to maintain a balance between the supervised and unsupervised component within the optimization process. α is proportional to the rate N/M where M denotes the number of labeled patterns. The estimations of cluster centers (prototypes) and the fuzzy covariance matrices are

$$v_s = \sum_{k=1}^{N} u_{sk}^2 x_k \Big/ \sum_{k=1}^{N} u_{sk}^2 \tag{4}$$

and

$$W_s^{-1} = \left[\frac{1}{\rho_s \det(P_s)} \right]^{1/n} P_s \tag{5}$$

respectively, where s =1, 2, …, c , ρ_s = 1 (all clusters have the same size), and

$$P_s = \frac{\sum_{k=1}^{N} u_{sk}^2 (x_k - v_s)(x_k - v_s)^T}{\sum_{k=1}^{N} u_{sk}^2}, s = 1,2,\cdots,c \tag{6}$$

The Lagrange multiplier technique yields an expression for partition matrix

$$u_{st} = \frac{1}{1+\alpha}\left\{ \frac{1+\alpha\left(1-\sum_{j=1}^{c} f_{jt}\right)}{\sum_{j=1}^{c} \frac{d_{st}^2}{d_{jt}^2}} + \alpha\left(f_{st} b_t\right) \right\} \quad s = 1,2,\cdots,c, \quad t = 1,2,\cdots,N \tag{7}$$

Using an alternating optimization (AO) method, the SSFCM algorithm iteratively updates the cluster centers, the fuzzy covariance matrices and the partition matrix by (4), (5) and (7) respectively until some termination criteria are satisfied.

3.2.2. Proposed semi-supervised fuzzy clustering method for class distribution learning

We first pre-process the retrieval experience using the following rules ($i = 1, 2, \ldots, c$ and $k = 1, 2, \ldots, N$) ——

(i) If $p_{ik} >> q_{ik}$, we can conclude that image k should be ascribed into the ith cluster, i.e., u_{ik} should be large compared to other u_{jk} ($j = 1, 2, \ldots, c, j \neq i$);

(ii) If $p_{ik} << q_{ik}$, image k should not be ascribed into the ith cluster, i.e., u_{ik} should be close to zero;

(iii) If (i) and (ii) are not satisfied, we cannot make any conclusion on ascribing image k ($k = 1, 2, \ldots, N$) into the ith cluster, i.e., we have no idea on the value of u_{ik} so we have to execute fuzzy clustering to derive its value.

Following the above discussion, we construct two new matrixes $\Phi_{c\times N}$ and $\Psi_{c\times N}$, the first of which represents positive information while the latter represents the negative information. For element ϕ_{ik} of Φ, if p_{ik} and q_{ik} satisfy Condition (i), $\phi_{ik} = 1$; otherwise, $\phi_{ik} = 0$. For element ψ_{ik} of Ψ, if p_{ik} and q_{ik} satisfy Condition (ii), $\psi_{ik} = 1$; otherwise, $\psi_{ik} = 0$.

We then normalize non-zero columns of P, namely, if $\sum_{i=1}^{c} p_{ik} > 0$, then $p_{jk} = p_{jk} \Big/ \sum_{i=1}^{c} p_{ik}$, $j = 1, 2, \ldots, c,$ $k =$ 1, 2, …, N. The purpose of normalization is to estimate the membership grades of the marked images.

Our objective function is similar to that in (3) with the modification

$$J_2 = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2 d_{ik}^2 + \alpha \sum_{i=1}^{c}\sum_{k=1}^{N} (u_{ik} - p_{ik})^2 d_{ik}^2 \tag{8}$$

The task is to minimize the objective function J_2 with respect to the partition matrix and the prototypes of the clusters, namely $\min_{v_1, v_2, \ldots, v_c, U} J_2$ with respect to cluster centers $v_1, v_2, \ldots, v_c$ and U satisfying conditions (a) and (b) for fuzzy clustering and a new constraint: $u_{ik} = 0$ if $\psi_{ik} = 1$, $i = 1, 2, \ldots, c,$ $k = 1, 2, \ldots, N$. This new constraint implies that if we already know that a pattern should not be ascribed to a certain class, we can pre-define the corresponding

membership element to be zero. For the kth column of $\Psi_{c\times N}$, there are $n(k)$ non-zero elements, whose row indices are $I(k) = \{r_{1,k}, r_{2,k}, ..., r_{n(k),k}\}$. All other notations are the same as those in the first part of this section.

Using the technique of Lagrange multipliers, the optimization problem in (8) with constraints (a) and (b) for the fuzzy clustering, it is converted into the form of unconstrained minimization

$$J_2 = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2 d_{ik}^2 + \alpha \sum_{i=1}^{c}\sum_{k=1}^{N} (u_{ik} - p_{ik})^2 d_{ik}^2 - \sum_{k=1}^{N} \lambda_k \left(\sum_{i=1}^{c} u_{ik} - 1 \right) \tag{9}$$

From the optimization requirement $\frac{\partial J_2}{\partial u_{st}} = 0$, we get $u_{st} = \frac{1}{1+\alpha}\left(\frac{\lambda_t}{2d_{st}^2} + \alpha p_{st} \right)$, if $\psi_{st} = 0$; otherwise, $u_{st} = 0$. From the fact that the sum of the membership values, $\sum_{j=1}^{c} u_{jt} = 1$, we have $\frac{1}{1+\alpha}\left\{ \frac{\lambda_t}{2} \sum_{j=1, j\notin I(t)}^{c} \frac{1}{d_{jt}^2} + \alpha \sum_{j=1, j\notin I(t)}^{c} p_{jt} \right\} = 1$. So we get

$$u_{st} = \frac{1}{1+\alpha}\left\{ \frac{1+\alpha-\alpha \sum_{j=1, j\notin I(t)}^{c} p_{jt}}{\sum_{j=1, j\notin I(t)}^{c} \frac{d_{st}^2}{d_{jt}^2}} + \alpha p_{st} \right\} \tag{10}$$

The expressions of cluster centers and the fuzzy covariance matrices are the same as in (4) and (5) respectively. Our semi-supervised fuzzy clustering algorithm for learning class distribution is outlined in Fig. 3.

3.3. Incorporating meta knowledge into feature relevance learning

A kind of probabilistic feature relevance learning (PFRL) based on user's feedback, that is highly adaptive to query locations is suggested in (Peng *et al*., 1999). The main idea is that feature weights are derived from probabilistic feature relevance on a specific query (local dependence), but weights are associated with features only. Fig. 4 illustrates the cases at points near decision boundary where the nearest neighbor region is elongated in the direction parallel to decision boundary and shrunk in the direction orthogonal to boundary. This implies that the feature with direction orthogonal to the decision boundary is more important. This idea is actually the adaptive version of nearest neighbor technique developed in (Hastie and Tibshirani, 1996).

3.3.1. Proposed strategy for relevance feedback with fuzzy clustering

Using fuzzy clustering, we already get class distribution knowledge, which is represented by the partition matrix $U_{c\times N}$. We now transform this meta knowledge into *defuzzied partition matrix* $Z_{c\times N}$, i.e., update the elements of U by binary scale $\{0, 1\}$. The elements of $Z_{c\times N}$ are defined as: If $u_{ik} \geq \beta\left(\max_{j=1,2,\cdots,c} u_{jk}\right)$, z_{ik} =1; else, $z_{ik} = 0$, $i = 1, 2, \ldots, c$, $k = 1, 2, \ldots, N$. The value of $\beta \in (0, 1]$ represents to what extent we can say that the element u_{ik} is large enough so that image k can be ascribed to class i.

At any iteration, if M images ($I_1, I_2, \ldots, I_M$) are marked positive by the current user, we then check if these positive images can be ascribed into one common class. If $\exists\, s \in \{1, 2, \ldots, c\}$, $\forall\, k \in \{ I_1, I_2, \ldots, I_M\}$ that $z_{sk} = 1$, then the current user seems to be seeking the concept corresponding to class s. So the system can save the tremendous amount of work for feature relevance learning and searching K images over the entire database; Instead, only searching K images within class s is needed, i.e., searching among the images whose sth element of the corresponding U column vectors are 1.

When enough retrievals on the image database are executed by different users, the class distribution knowledge will be close to most human users concepts. This leads to not only saving computational time for retrieval, but also to improved retrieval precision.

4. Improved concept learning approach

For the approach presented in the previous section, the retrieval experience is directly represented by positive matrix P and negative matrix Q. How can the system derive such matrices? During the long-term learning process, each time after the current user ended his (her) query session, the system gets some positive images and some negative images from this user's feedback. Obviously, "positive" ("negative") means that the corresponding images do (do not) contain the concept the user has sought. If the system knows which concept is sought by the user, it can update P by locating those matrix elements corresponding to positive images and this concept and increasing the elements by 1; it can also update Q in a similar manner with respect to negative images.

Unfortunately, the system is not directly given which concept the user has sought. We have to use some technique to estimate the concept sought by the user so that P and Q are derived to help concept learning. This is the major

improvement of the concept learning approach presented in this section (we call it Algorithm B). We also modify fuzzy clustering algorithm and the strategy for relevance feedback.

4.1. Concept learning

After a user's retrieval experience, let there be N^+ positive labeled images and N^- negative labeled images, and they are represented by $\mathbf{I}^+ = \{I_1^+, I_2^+, \ldots, I_{N^+}^+\}$ and $\mathbf{I}^- = \{I_1^-, I_2^-, \ldots, I_{N^-}^-\}$ respectively. The task is to first determine which concept the user was seeking so that we can derive correct knowledge from this retrieval and then improve our concept learning by semi-supervised fuzzy clustering later. The index κ of the cluster corresponding to the concept sought is computed as

$$\kappa = \arg\max_{k=1,2,\ldots,c} P(k) \tag{11}$$

where $P(k)$ is equal to

$$\begin{aligned} &\Pr(I_1^+ \in C_k, \cdots, I_1^+ \in C_k, I_1^- \notin C_k, \cdots, I_1^- \notin C_k) \\ &= \prod_{i=1}^{N^+} \Pr(I_i^+ \in C_k) \prod_{j=1}^{N^-} \Pr(I_j^- \in C_k) \\ &= \prod_{i=1}^{N^+} u_{k,I_i^+} \prod_{j=1}^{N^-} (1 - u_{k,I_j^-}) \end{aligned} \tag{12}$$

with u_{kj} ($k = 1, 2, \ldots, c$ and $j = 1, 2, \ldots, N$) being the element of partition matrix $U_{c\times N}$ and C_k ($k = 1, 2, \ldots, c$) being concept k. This probability based maximization method uses the current partition matrix information to decide the sought concept, which necessitates the assumption that current partitioning is not too bad.

Now the images in $\mathbf{I}^+$ are in cluster κ and those in $\mathbf{I}^-$ are not in cluster κ. We designate the positive matrix $P_{c\times N}$ and the negative matrix $Q_{c\times N}$ to represent this kind of knowledge. At the very beginning, when no retrieval has ever been executed on the system, P and Q are initialized to be zero matrices. After a retrieval experience, the elements $\{p_{\kappa,I_1^+}, \ldots, p_{\kappa,I_{N^+}^+}\}$ in P and the elements $\{q_{\kappa,I_1^-}, \ldots, q_{\kappa,I_{N^-}^-}\}$ in Q are increased by 1. So the values of $p_{\kappa j}$ and $q_{\kappa j}$ represent to what extent people agree and disagree to ascribe an image j into cluster κ, respectively.

The motivation for having matrices P and Q is to capture and update previous users' retrieval experiences. In the following, P and Q are processed in the sense of statistics by estimating users' voting whether a certain image contains a specific concept or not.

Define $E = P - Q$, and let $b_j = 0$, if the jth column in E is a zero vector; 1, otherwise. Let M be the number of normalized columns of E, we define $\alpha = N/M$. We then let F be the matrix that has normalized columns of E, i.e., for the elements of F,

$$f_{kj} = \frac{e_{kj} - \min\limits_{i=1,2,\ldots,c} e_{ij}}{\max\limits_{i=1,2,\ldots,c} e_{ij} - \min\limits_{i=1,2,\ldots,c} e_{ij}} \tag{13}$$

for $k = 1, 2, \ldots, c, j = 1, 2, \ldots, N$ and kth column in E is a non-zero vector.

If the element e_{kj} of E is negative, $k = 1, 2, \ldots, c, j = 1, 2, \ldots, N$, it implies that there are fewer people ascribing image j to cluster k than those opposing to this association, we conclude that image j does not contain concept k and directly predefine the element u_{kj} of partition matrix to zero. If for the jth column of $E_{c\times N}$, there are l_j negative elements whose row indices are $J(j) = \{ r_{1,j}, r_{2,j}, \ldots, r_{l_j,j} \}$, we set $e_{kj} = 0, j = 1, 2, \ldots, N, k \in J(j)$.

We can now deal with the semi-supervised fuzzy clustering, which is also an optimization problem with the objective function (3). Besides the two constraints (a) and (b) appearing in 3.2, a new constraint is added as we have discussed above:

$$(c)\ u_{kj} = 0,\ j = 1, 2, \ldots, N,\ i \in J(k) \tag{14}$$

The estimations of cluster prototypes and the fuzzy covariance matrices are also (4) and (5) respectively. And we derive the expression for partition matrix elements as

$$u_{st} = \frac{1}{1+\alpha}\left\{ \frac{1+\alpha\left(1 - b_j \sum\limits_{j=1, j\notin J(t)}^{c} f_{jt}\right)}{\sum\limits_{j=1, j\notin J(t)}^{c} \frac{d_{st}^2}{d_{jt}^2}} + \alpha\left(f_{st} b_t\right) \right\} \tag{15}$$

where $s = 1, 2, \ldots, c$ and $t = 1, 2, \ldots, N$.

4.2. Improving retrieval performance

As introduced in 3.3.1, we first defuzzy the partition matrix $U_{c\times N}$ to $Z_{c\times N}$. With user's feedback after iteration 0, if L^+ images $\{I_1^+, I_2^+, \ldots, I_{L^+}^+\}$ are labeled positive and L^- images $\{I_1^-, I_2^-, \ldots, I_{L^-}^-\}$ are labeled negative by user, we check if these positive images can be ascribed into one common cluster while negative images are not in this cluster. If $\exists\ s \in \{1, 2, \ldots, c\}$, the following two conditions are satisfied:

(a) $\forall\, j \in \{I_1^+, I_2^+, \ldots, I_{L^+}^+\}, z_{sj} = 1,$

(b) $\forall\, i \in \{I_1^-, I_2^-, \ldots, I_{L^-}^-\}, z_{si} = 0,$

then the current user seems to be seeking the concept corresponding to cluster *s*. So the system saves tremendous amount of computation for feature relevance learning and searching *K* images over the entire database; instead, only searching *K* images within cluster *s* is needed, i.e., searching among the images whose *s*th element of the corresponding *U* column vectors are 1. When above conditions are not satisfied, we use statistical feature relevance approach presented in {Peng, *et al*., 1999} to perform the retrievals and update clustering.

Our concept learning algorithm with fuzzy clustering and relevance feedback is outlined in Fig. 5.

5. Experiments

We first present experimental results on both synthetic data and real data using the approach introduced in Section 3. Then we demonstrate the improved approach in Section 4 on synthetic and real data.

To evaluate the result of fuzzy clustering, we define the *groundtruth matrix* $G_{c\times N}$, whose element g_{ij} ($i = 1, 2, \ldots, c$ and $j = 1, 2, \ldots, N$) is defined as: $g_{ij} = 1$, if image *j* has concept *i*; 0, otherwise.

An important measure for the fuzzy clustering result is the *percentage of correct clustering*, which is defined as

$$\text{percentage} = \frac{\sum_i \sum_j g_{ij} \cdot xor \cdot z_{ij}}{cN} \tag{16}$$

where z_{ij} is the element of defuzzied partition matrix *Z* as defined in Section 3.3.1.

The retrieval performance is measured by *precision*, which is defined as

$$\text{precision} = \frac{\text{number of positive retrievals}}{\text{number of total retrievals}} \times 100\% \tag{17}$$

5.1. Synthetic data —— Algorithm A

Fig. 6 shows a synthetically created two-dimensional pattern. It consists of three overlapping clusters: two of them are ellipsoidal (class 1 and class 2) while the third one (class 3) is a circle. The two ellipsoidal clusters have the same means $[\ 0\ 0\ ]^T$, and their covariance matrix given as rows are [12 -6.8 ; -6.8 4] and [12 6.8 ; 6.8 4] respectively. The third cluster has mean of $[\ -1\ 0]^T$ and its covariance matrix is [1 0 ; 0 1]. The size of each cluster is 50, so we

have 150 patterns in total. For standard fuzzy clustering, the correct percentage is only 36.7%, which is close to the guess value 1/3. This is not unusual because clusters significantly overlap.

We then test both Pedrycz's clustering algorithm (Pedrycz and Waletzky, 1997) and our algorithm on this data with different amounts of experience. Experience is defined as the ratio of the number of labeled patterns to the total number of patterns. When the experience is γ, we randomly choose γN patterns and label them positive for their groundtruth clusters; at the same time, randomly choose γN patterns, and for each pattern, label it negative for one cluster that is not its groundtruth cluster. Then repeat clustering with respect to this experience 10 times, and calculate the average correct percentage. For Pedrycz's method, only positive experience is used while for our method both positive and negative experiences are used. Fig. 7 shows that with increasing experience, the percentage of correct clustering becomes better and that the result of our method is better than Pedrycz's. Fig. 8 shows the misclassified patterns by our method with respect to different experience values. This shows the advantage of our algorithm for learning high-level concepts since in addition to positive feedback, negative feedback is also available from user's responses.

5.2. Real data —— Algorithm A

We construct two image databases with sizes of 180 and 1047 respectively for experiments.

5. 2. 1. Database I

This image database consists of a variety of images all containing one or more of the following five objects: *water*, *sun*, *sky*, *cloud* and *ground*. The total number of images is 180. Each image is annotated with five labels (0 or 1), so the groundtruth class distribution can be represented by a matrix $G_{180\times5}$ whose elements are 0-1 value. Fig. 9 shows sample images. The numbers of images within the five classes are 49, 63, 83, 130, 59, respectively. Each image in this database is represented by 16-dimensional feature vectors obtained using 16 Gabor filters for feature extraction (Peng *et al.*, 1999).

Our semi-supervised fuzzy clustering algorithm is applied to the data with different amounts of experience, N = 180, c = 5, K = 16, α = 1, β = 0.5. Fig. 10 shows the percentage of correct clustering with respect to different experience. The percentage of correct clustering is determined by comparing the elements of the groundtruth matrix G and those of defuzzied partition matrix U.

We then randomly select one of the 180 images as query, and other 179 remaining images as training samples. The retrieval process is automatically executed since we use the groundtruth matrix $G_{180\times5}$ to provide user's interactions: At first, randomly select a concept that the query image can be ascribed to, and regard this concept as what the user is seeking. When the retrieval system presents the resulting K images, we use matrix $G_{180\times5}$ to mark them. If the membership element of the $G_{180\times5}$ corresponding to the image with respect to desired concept is 1, then mark this image positive; otherwise, it is marked as negative. By repeating such retrievals 50 times by selecting a different image as query each time, we obtain the average precision results shown in Fig. 11.

We observe that when only PFRL is used, the average precision (= 58.1%) is the lowest. With the increasing experience, the average precision becomes higher. Experience of 10% helps to increase the precision significantly (precision = 68.9%). When the experience is 20%, the precision reaches 88.0%. These results support the efficacy of our method.

Fig. 12 and Fig. 13 show four groups of sample retrievals in total when 20% experience is available. The query image in each group contains different number of concepts from 1 to 4. The retrieval results at the second iterations are improved over those at the first iterations with the help of meta knowledge derived from the experience using fuzzy clustering. For example, the query image in Fig. 12 (b) contains two concepts: *cloud* and *ground*. The user is seeking the concept *cloud.* At the first iteration, the system makes K-nearest neighbor search and only 5 out of the 16 resulting images contain *cloud*. At the second iteration, the system incorporates the class distribution knowledge into relevance feedback framework and 14 out of 16 images contain *cloud*.

5. 2. 2. Database II

This database contains 1047 images, which includes all the images in Database I. There are 9 concepts (of sizes): *plant* (115), *sky* (128), *animal* (100), *sunset* (199), *building* (249), *texture* (152), *people* (185), *cloud* (204) and *water* (146). On the average, each image contains 1.41 concepts. Besides the 16 texture features used in Database I, we also extract means and standard deviations from the three channels in HSV color space. Thus, each image is represented by 22 features.

We implement our fuzzy clustering method on this database, with $c = 9$, $K = 16$, $N = 1047$, $\alpha = 1$ and $\beta = 0.5$. Fig. 14 shows the percentage of correct clustering with respect to different experience. Observe that with experience

increased, the percentage of correct clustering is improved. The average precision results with different experience are shown in Fig. 15.

Fig. 16 shows two groups of sample retrievals when 80% experience is available. In (a), the user is seeking *cloud*. The *K*-NN search at the first iteration only gives 9 *cloud* images. At the second iteration, the class distribution knowledge help to give 16 *cloud* images. In (b), the user is seeking *water*, 8 *water* images are given at the first iteration and 14 *wate*r images are given at the second iteration.

5. 3. Experimental results using the improved approach

For each retrieval, the user's interaction is monitored by the groundtruth matrix $G_{c\times N}$.

5. 3. 1 Synthetic data —— Algorithm B

Fig. 17 shows three synthetically created overlapping clusters (two-dimensional, Gaussian distribution). Each cluster contains 50 patterns. Cluster 1 and Cluster 2 are ellipsis with the same mean of $[\ 0\ 0\ ]^T$ and they have covariance matrices (given as rows) [3.0625 –1.6238; -1.6238 1.1875] and [3.0625 1.6238; 1.6238 1.1875] respectively. Cluster 3 is a circle with the mean of $[\ -1\ 0\ ]^T$ and covariance matrix (given as rows) [1 0; 0 1]. Fig. 17 (a) shows the cluster distribution.

We implement our clustering algorithm on this synthetic data with $c = 3$, $N = 150$, $K = 8$, and $\beta = 1$. Simulating the system with increased retrieval experiences (the number of users' retrieval sessions), we randomly select a pattern as the query for each retrieval, and decide the concept (cluster) that is sought by positive and negative images. We then update the fuzzy clustering and derive the defuzzied partition matrix. An example of this process is shown in Fig. 17 (b-d), in which the clustering result is improved with increased experiences.

Fig. 18 shows the average percentage of correct clustering with increased experiences. Notice that only 89.7% of correct clustering is achieved after 100 experiences. This is because the partition matrix derived from the initial fuzzy c-means clustering without any experience is far away from groundtruth matrix. After a user's experience, the system may mistakenly decide the concept sought. This incorrect knowledge will mislead the fuzzy clustering which may cause the updated partition matrix to be farther away from groundtruth matrix. After a retrieval experience, if the correctly sought concept is directly given instead of deriving it by computation, this is called a training experience. Fig. 18 also gives the performance curve with training experiences, which help clustering result to finally reach 100%. The role of training stage will be discussed further in the real data experiment.

5. 3. 2 Real data —— Algorithm B

In this section, we implement the improved algorithm on **Database II** introduced in Section 5.2.2. We simulate the process of a retrieval system for which queries are selected randomly among the patterns in the database.

We implement our fuzzy clustering method on this database, with c = 9, N = 1047, K=16 and β = 0.5. For the reasons of the big gap between low-level features and a human concept, the initial fuzzy clustering is far away from groundtruth labeling. We can set a training stage at the beginning of the system's running online. Let there be t training experiences, in each of which on the average L images are labeled positive or negative, the amount of concept knowledge derived from training is estimated to be $\frac{tL}{cN}$, which denotes the percentage of elements whose values are given in advance out of all the elements in the groundtruth matrix.

Fig. 19 shows the fuzzy clustering performance of the system going through 500 retrieval experiences starting with different amounts of training experiences. With increased number of initial training experiences, fuzzy clustering is improved. Compared with the case that has no training, 20 training experiences improve the clustering significantly. In our experiment, L = 26, so the amount of concept knowledge derived from the 20 training experiences is 5.5% . We also observe from Fig. 19 that even with training experiences, the percentage of correct clustering still cannot converge to 100%, which again reflects the gap between image features and human visual concepts.

For concept k, k = 1, 2, …, c, in the corresponding kth rows in groundtruth matrix G and defuzzied partition matrix Z, for j = 1, 2, …, N, let

$N1$ = number of j that give $g_{kj} = 1$,

μ = number of j that give $g_{kj} = 1$ and z_{kj}=0,

$N0$ = number of j that give $g_{kj} = 0$,

ν = number of j that give $g_{kj} = 0$ and $z_{kj} = 1$.

We define the Probability of detection and Probability of false alarms as $Pd = (N1 - \mu)/N1$ and $Pf = \nu/N0$. Calculating the average Pd and Pf over the c concepts, we obtain the ROC curves for detection performance of partition matrix with different amounts of experiences shown in Fig. 20. With the value of defuzzy parameter β

decreased, *Pd* and *Pf* both becomes larger. Observe that with more retrieval experiences, in the case when β is not very large, the detection ability of partition matrix is improved.

Fig. 21 presents the retrieval performances with different amounts of experiences starting with 20 training experiences. We select an image in this database as the query, implement our retrieval strategy, and repeat this experiment by changing query until each of the 1047 images has been selected as query. Then we calculate the average precision at each iteration. Among these 1047 queries, the number of those leading to direct search within a cluster is 174, 289 and 421, respectively corresponding to 200, 300 and 500 experiences. If the percentage of correct clustering is high, the retrieval with direct search within a cluster yields a high precision after iteration 0, so it is not strange that with increased experiences, the average retrieval precision is improved. The more important aspect of direct search within one cluster is that the computational time at iteration 1 is decreased by 1/*c* compared with that of searching the entire database. This has deep significance for retrieval performance in practical applications. Fig. 22 shows two different retrievals with the same query image which is regarded as containing the concepts of both *cloud* and *water* based on the concept learning after 500 experiences.

5. 4. Discussions

Since real image database is incrementally changed with addition or removal of images from the database, the size of partition matrix U changes correspondingly. In the following, we consider the two cases of image addition and removal separately. Let the current size of database be N_0, and the current partition matrix be U_0, whose size is $c \times N_0$. When a new image is added, the size of partition matrix U becomes $c \times (N_0+1)$, and the (N_0+1)th column corresponds to the new image. When the fuzzy clustering is to be implemented on the database again after the new retrieval experience is obtained, in partition matrix, the initial values of the elements corresponding to the original N images are set to be those in U_0, and the elements corresponding to the new image are randomly initialized with the constraint that the summation of these elements be 1. Since neither the feature vector of a new image nor a new retrieval experience will change the clustering significantly, only a few iterations of updating are needed. Similarly, when some images are removed from the database, to partition the remaining images, the initial partition matrix element values are correspondingly set to be those in U_0.

As for computational load of the clustering algorithm, since inverse matrix computation is required at each iteration, it is obviously not fast. Fortunately, when a new query comes, to present images to user, the system does not have to implement the on-line clustering, instead, the system only needs to use existing clustering result to help

relevance feedback. There may be a time lag for the on-line clustering due to its computational complexity. For example, when $\mathcal{N}$th query comes, the system may only finish clustering based on retrievals 1 to ($\mathcal{N}$ - τ), where τ is very small compared with $\mathcal{N}$. The system will use this clustering result to help relevance feedback during the $\mathcal{N}$*th* retrieval. Since there is little information difference between retrieval 1 to ($\mathcal{N}$ - τ) and retrievals 1 to ($\mathcal{N}$ - 1), the retrieval performance is barely influenced by this on-line clustering time lag. From the above observation, the clustering lag has little influence on retrieval performance so long as the clustering time is far below the average retrieval time (frequency), which is generally satisfied in real image databases. For this reason, computational load of the clustering is not our main concern in this paper.

6. Conclusions

This paper presented two approaches for incorporating meta knowledge into the relevance feedback framework to improve image retrieval performance. We first give Algorithm A based under the assumption that the retrieval experience is directly represented by positive matrix and negative matrix. Algorithm B derives retrieval experience by using a probabilistic technique and modifies concept learning and relevance feedback. We find that Algorithm B is promising for concept learning. The modified semi-supervised fuzzy clustering method can effectively learn class distribution in the sense of high-level concept from retrieval experience. Using fuzzy rules, we adapted the meta knowledge into relevance feedback to improve the retrieval performance. With more retrievals on the image database by different users, the class distribution knowledge became closer to typical human concepts. This leads faster retrieval with improved precision. The consequence of this is to be able to handle more effectively a large database. In the future, we plan to show results on a larger and more complex image database. The dynamic concept creation, splitting and merging are also the topics of future research.

Acknowledgements

This work was supported by DARPA/AFOSR grant F49620-97-1-0184. The contents of the information do not necessarily reflect the position or the policy of the US Government.

References

Bensaid, A. M., Hall, L. O., Bezdek, J. C., Clarke, L. P., 1996. Partially supervised clustering for image segmentation. Pattern Recognition 29 (5), 859-871.

Bezdek, J. C., Keller, J., Krisnapuram, R., Pal, N. R., 1999. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Kluwer Academic Publisher, Boston.

Bhanu. B., Dong, A., 2001. Exploitation of meta knowledge for learning visual concepts. Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries, Kauai, HI, 81-88.

Chang, S. F., Chen, W., Sundrarm, H., 1998. Semantic visual templates - linking visual features to semantics. Proc. IEEE Int. Conf. Image Processing (ICIP'98) 3, Chicago, IL, 531-535.

Cox, I. J., Miller, M. L., Minka, T. P., Papathomas, T. V., Yianilos, P.N., 2000. The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. IEEE Trans. on Image Processing 9 (1), 20-37.

Flickner, M., *et al*., Query by image and video content: the QBIC system. IEEE Computer 28 (9), 23-32.

Gustafson, D. E., Kessel, W. C., 1978. Fuzzy clustering with a fuzzy covariance matrix. Proc. IEEE Conf. on Decision and Control, San Diego, CA, 761-766.

Hastie, T., Tibshirani, R., 1996. Discriminant adaptive nearest neighbor classification. IEEE Trans. on Pattern Analysis and Machine Intelligence 18 (6), 607-616.

Jain, A., Murty, M., Flynn, P., 1999. Data clustering: a review. ACM Computing Surveys 31 (3), 264-323.

Lim, J. H., 1999. Learning visual keywords for content-based retrieval. Proc. IEEE Int. Conf. Multimedia Computing and Systems (ICMCS'99) 2, Florence, Italy, 169-173.

Lipson, P., Grimson, E., Sinha, P., 1997. Configuration based scene classification and image indexing. Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'97), San Juan, Puerto Rico, 1007-1013.

Minka, T., Picard, R., 1997. Interactive learning with a society of models. Pattern Recognition 30 (4), 565-581.

Pedrycz, W., Waletzky, J., 1998. Fuzzy clustering with partial supervision. IEEE Trans. on Systems, Man, and Cybernetics 27 (5), 787-795.

Peng, J., Bhanu, B., Qing, S., 1999. Probabilistic feature relevance learning for content-based image retrieval. Computer Vision and Image Understanding 75 (1-2), 150 -164.

Ratan, A. L., Maron, O., Grimson, W. E. L., Lozano-Perez, T., 1999. A framework for learning query concepts in image classification. Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'99) 1, Fort Collins, CO, 423-429.

Rui, Y., Huang, T., 2000. Optimizing learning in image retrieval. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'00), Hilton Head Island, SC, 236-243.

Rui, Y., Huang, T., Ortega, M., Mehrotra, M., 1998. Relevance feedback: a power tool for interactive content-based image retrieval. IEEE Trans. on Circuits and Systems for Video Technology 8 (5), 644-655.

Tieu, K., Viola, P., 2000. Boosting image retrieval. Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'00), Hilton Head Island, SC, 228-235.

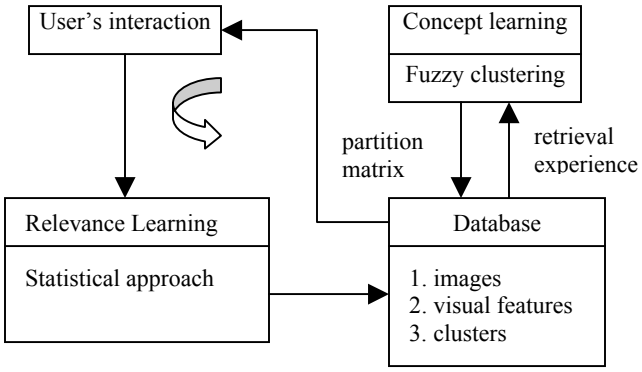


Fig. 1. Simplified system diagram for concept learning using meta knowledge.

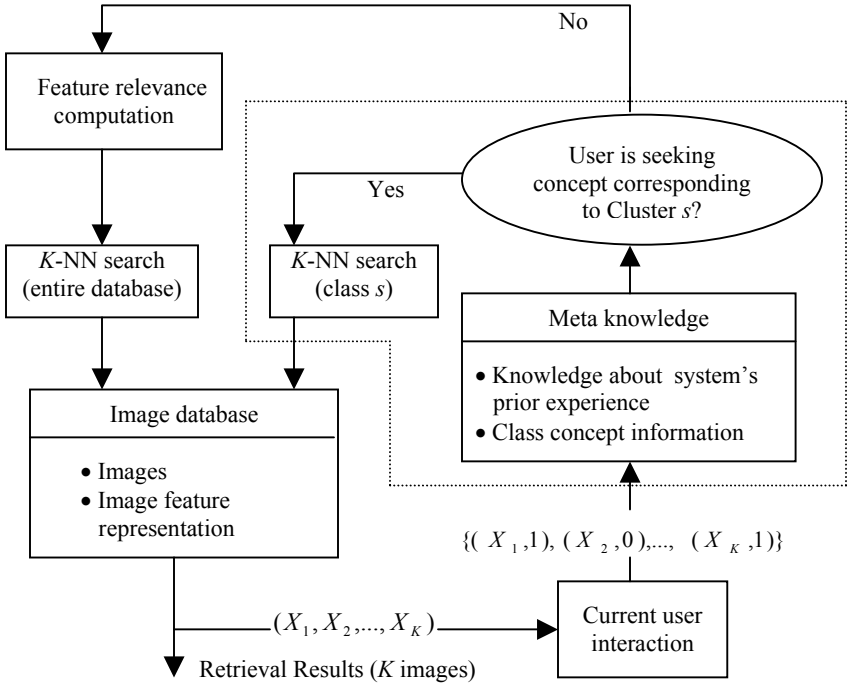


Fig. 2. Detailed system diagram for concept learning using meta knowledge.

1. Given the number of clusters c, positive matrix P, negative matrix Q . Select the distance function as Euclidean distance.
2. Compute new matrices $\Phi_{c\times N}$ and $\Psi_{c\times N}$. Initialize partition matrix U: If $\psi_{ik} = 1$, $u_{ik} = 0$; Otherwise, set u_{ik} randomly in the interval [0, 1] so that the sum of each column of U is 1.
3. Compute cluster centers and the fuzzy covariance matrices by (4) and (5).
4. Update partition matrix: If $\psi_{ik} = 1$, $u_{ik} = 0$; Otherwise, compute the element by (10).
5. If $\left\|U - U^{'}\right\| < \delta$ (with δ being a tolerance limit) then stop, else go to 3 with $U = U^{'}$.

Fig. 3. Algorithm A — **s**emi-supervised fuzzy clustering algorithm (SSFCM) for concept learning.

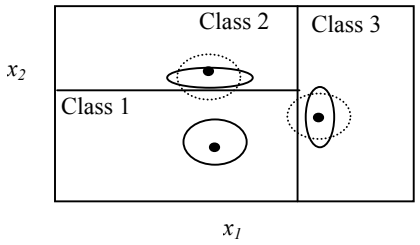


Fig. 4. Feature weights are different along different dimensions. The dotted circles represent the equally likely nearest neighborhood and the solid ellipses represent feature-weighted nearest neighborhood.

Given the number of clusters c, the number of images N.
Initialize positive matrix $P_{c\times N}$ and negative matrix $Q_{c\times N}$ to be zero matrices.
Repeat
A user starts his (her) retrieval session by inputting a query image;
flag $\leftarrow$ 1;
While (flag = 1)
If the system can decide that user is seeking a concept corresponding to Cluster s
Search images within Cluster s;
flag $\leftarrow$ 0;
Else
Probabilistic Feature Relevance Learning (PFRL);
End if
End while
If (flag = 0)
1. Compute κ by (11) and update P and Q, then compute matrix F and α;
2. Compute cluster centers and the fuzzy covariance matrices by (4) and (5);
3. Update partition matrix: if not predefined as 0, the elements are computed by (15);
4. If $\|U - U'\| < \delta$ (with δ being a tolerance limit), stop; else, go to 2 with $U = U'$;

End if

Fig. 5. Algorithm B — concept learning with fuzzy clustering and relevance feedback.

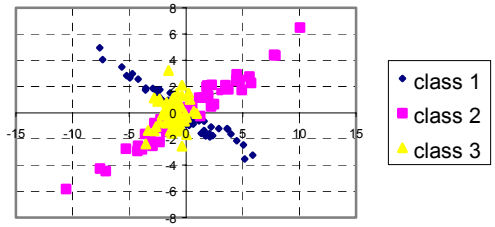


Fig. 6. Two-dimensional data distribution with three overlapping clusters.

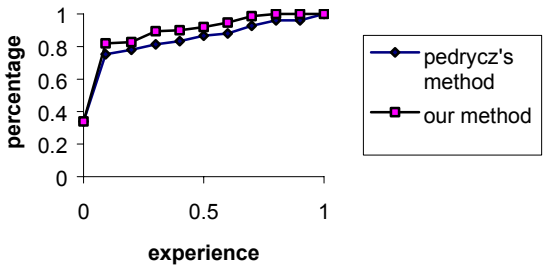


Fig. 7. Clustering results by two methods with different experience.

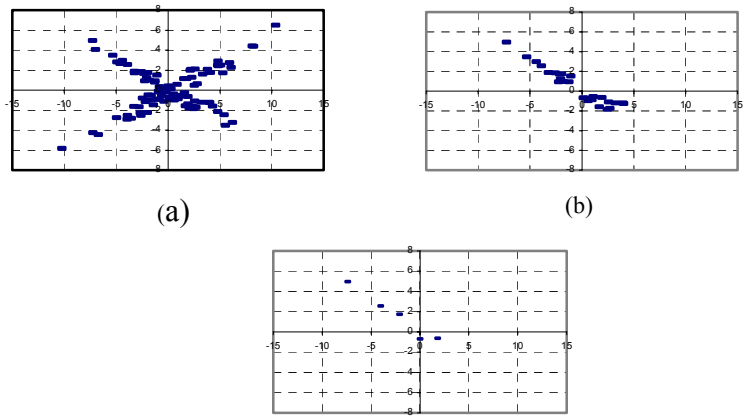


Fig. 8. Misclassified patterns for synthetic data set: (a) no experience, (b) 20% experience, (c) 50% experience.

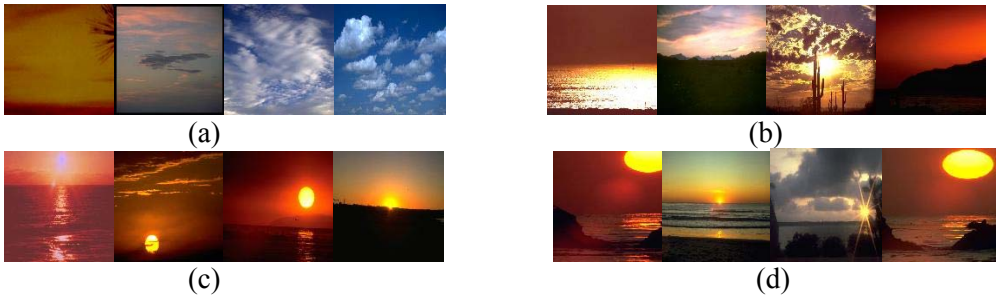
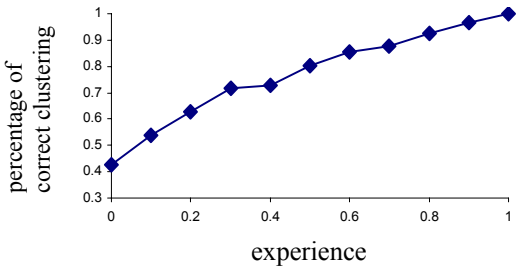


Fig. 9. Sample Images from real-world database: (a) images having one concept; (b) images having two concepts; (c) images having three concepts; (d) images having four concepts.



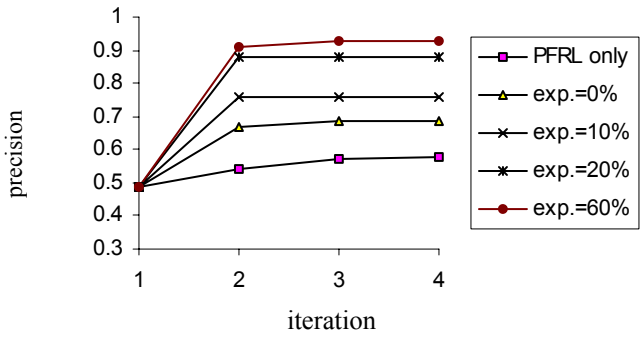
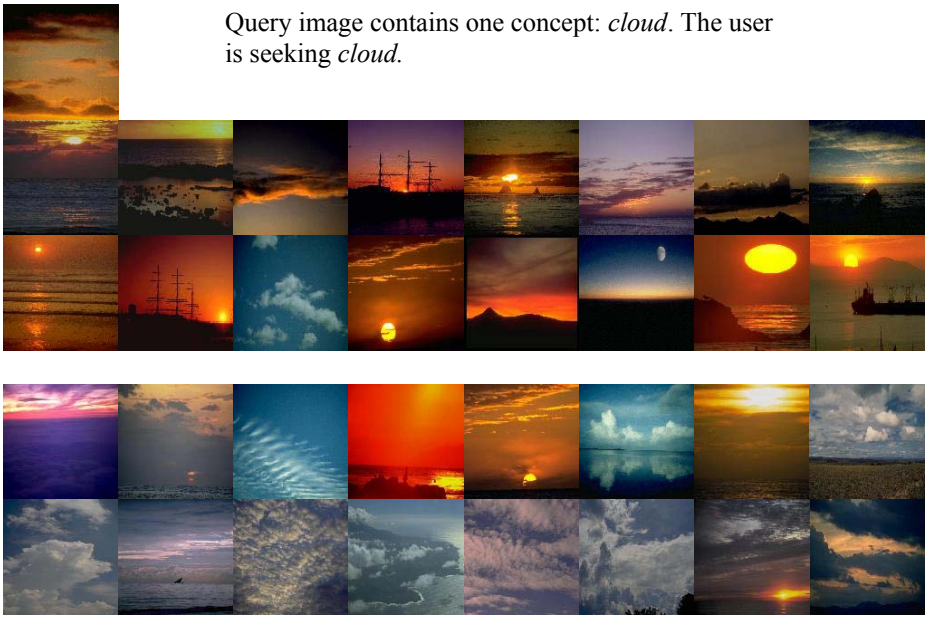


Fig. 11. Retrieval precisions for different experience.



(a)

(b)

Fig. 12. The sample (top 16) retrieval results (experience = 20%) at the first and the second iterations with query image containing (a) one concept, (b) two concepts.

(a)

(b)

Fig. 13. The sample (top 16) retrieval results (experience = 20%) at the first and the second iterations with query image containing (a) three concepts, (b) four concepts.

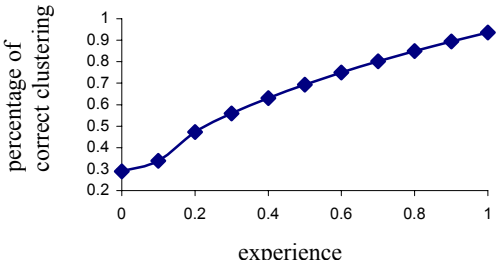


Fig. 14. Clustering results for real data with different experience.

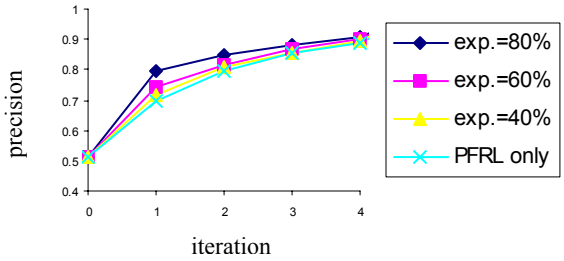
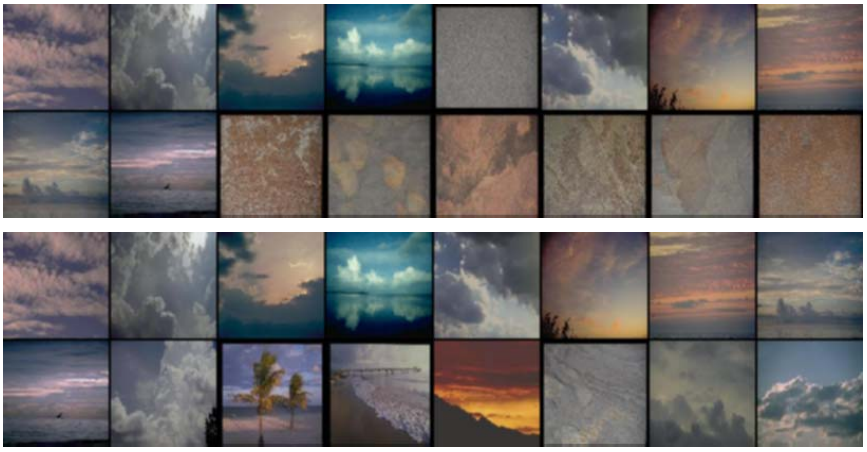


Fig. 15. Retrieval precisions for different experience.

Query image contains one concept: *cloud*. The user is seeking *cloud.*



(a)

Query image contains two concepts: *sunset* and *water*. The user is seeking *water.*

(b)

Fig. 16. The sample (top 16) retrieval results (experience = 80%) at the first and the second iterations with query image containing (a) one concept, (b) two concepts.

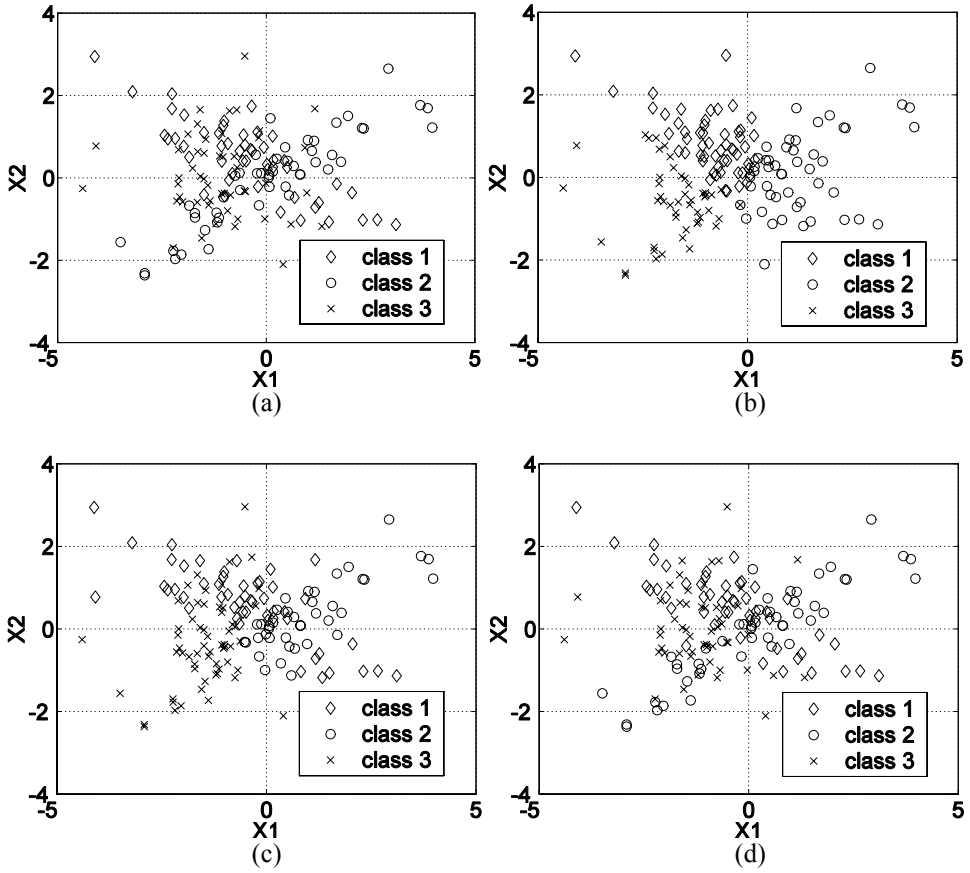


Fig. 17. Fuzzy clustering results. (a) groundtruth labels, (b) 0 experience (47 errors), (c) 10 retrievals (30 errors) and (d) 30 retrievals (5 errors).

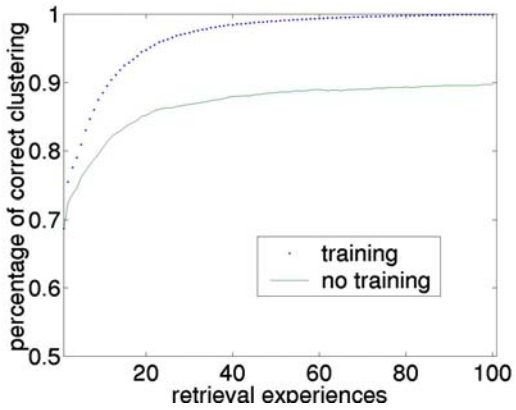


Fig. 18. Synthetic data: improved clustering with increased number of retrieval experiences.

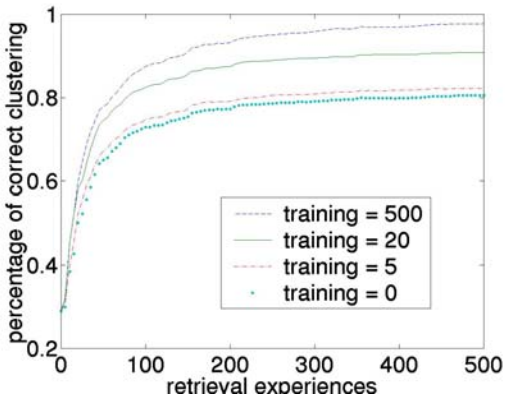


Fig. 19. Improved clustering with different amounts of training.

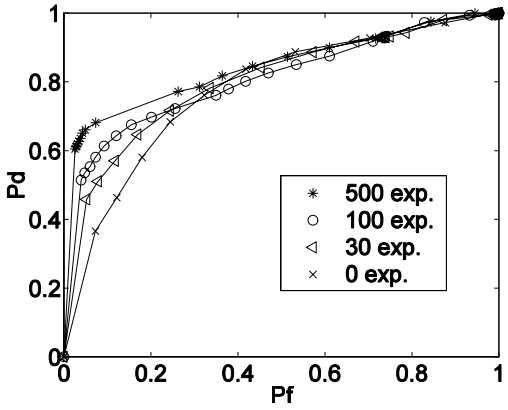


Fig. 20. ROC curves for database classification with different amounts of retrieval experiences.

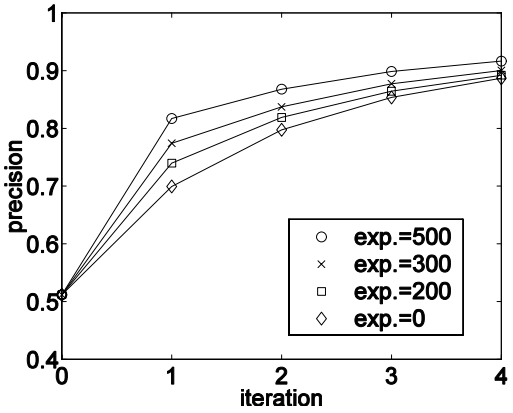


Fig. 21. Retrieval performance with various amounts of experiences.

(a)

(b)

(c)

Fig. 22. Different retrieval results with the same query (the first image) containing the concepts of *cloud* and *water*. The retrievals are shown after 500 experiences. Initially K-NN search yields the images in (a). When the user seeks *cloud*, 7 images having *cloud* are labeled positive (row 1: image 1, 6, 7; row 2: image 1, 3, 4, 5). After searching the *cloud* cluster, the retrieved images are shown in (b) with 12 correct images (except row 2: image 4, 5, 6, 8). When the user seeks *water*, 7 images in (a) are labeled positive (row 1: image 1, 4, 5, 7 and row 2: image 1, 2, 8). After searching the corresponding cluster, the retrieved images are shown in (c) with 15 correct images (except row 2: image 7).