Interactive Search for Image Categories by Mental Matching

Marin Ferecatu
IMEDIA Project, INRIA Rocquencourt
Marin.Ferecatu@inria.fr

Donald Geman
Dept. of Applied Mathematics and Statistics, The Johns Hopkins University
geman@jhu.edu

Abstract

Traditional image retrieval methods require a "query image" to initiate a search for members of an image category. However, when the image database is unstructured, and when the category is semantic and resides only in the mind of the user, there is no obvious way to begin (the "page zero" problem). We propose a new mathematical framework for relevance feedback based on mental matching and starting from a random sample of images. At each iteration the user declares which of several displayed images is closest to his category; performance is measured by the number of iterations necessary to display an instance. Our core contribution is a Bayesian formulation which scales to large databases with no semantic annotation. The two key components are a response model which accounts for the user's subjective perception of similarity and a display algorithm which seeks to maximize the flow of information. Experiments with real users and a database with 20,000 images demonstrate the efficiency of the search process.

1. Introduction

As the number of available multimedia documents steadily increases, so too does the need for efficient organization and retrieval of their content, which spurs research in content-based image retrieval [11]. The most popular methods today for searching for images residing in large unstructured repositories are query-by-example and interactive retrieval by relevance feedback. In the former case, the retrieved images express an overall visual similarity to a specified member of the database, the "query" image. Therefore, if the user's concept of similarity is largely thematic, then efficiency is adversely affected by the infamous "semantic gap"– the discrepancy between the representation of images in the database in terms of low-level features and the high-level, semantic descriptions meaningful to most users [8].

One strategy that has proven partially successful in practice for dealing with these differing representations is relevance feedback: solicit information from the user incrementally by dividing the search into consecutive rounds and allow the user to provide feedback at each step [16], for example by declaring some displayed images to be "relevant" or "similar" to a desired image category. In this way, the system can progressively refine a model of the user's target for the current search session.

Nonetheless, in order to begin a query session involving either query-by-example or relevance feedback, a starting example is needed; this is the "page zero problem." Traditionally, one simply displays randomly sampled pages from the image database until the user identifies an image of interest or a suitable starting point for directed search. Whereas this may be satisfactory for small databases, it rapidly becomes impractical for larger ones.

Various methods have been explored directly for the page zero problem, such as database categorization [14] and query construction [6]. Other methods, initially directed towards other objectives, such as mental matching for target search [3, 5] and automatic semantic annotation [2, 12], could be adapted to the initialization problem. In §2 we summarize the connections between these, and other articles, with our work.

Our main contribution is a new, iterative approach for discovering an instance from a semantic image category residing in the mind of the user. The search is terminated upon displaying one of these images and performance is measured by the expected number of iterations necessary to achieve this. No semantic annotation is assumed. Also, unlike previous approaches to mental matching, ours extends to category search and large, unstructured databases. It could serve either as a standalone function in a retrieval system or as a method for initializing another session, such as query-by-example, to obtain additional examples.

The core of our framework is a new statistical model for relevance feedback by mental matching. A binary random variable is assigned to every image in the database; the value is one if that image belongs belongs to the target class and zero if it does not. Hence, taken together, these variables determine the category. The relevance feedback session starts with a random screen. At each iteration, the user is asked to choose from among the displayed images the one that is closest to his target category using whatever criteria he desires. These decisions are inevitably subjective; indeed, the challenge is to design an answer model (i.e., a probability distribution for the user's response conditional on the membership status of any given image) which accounts for the nature of human decision-making, hopefully capturing the gap between the user's "metric" and the one used by the system.

The system maintains a separate, iteration-dependent posterior distribution for each image. Probabilities are updated based on the evidence gathered from the search, i.e., the responses of the user. Theoretically, the optimal new display would minimize the conditional entropy on the whole family of membership variables conditional on the search history and the new response. As this is computationally intractable, we use an extension of the heuristic proposed in [5], which is shown to work very well in practice. Moreover, in order to overcome certain problems introduced by the redundancy among images with very similar low-level descriptors, we use an unsupervised categorization of the database into small clusters that are visually highly coherent. The efficiency of the search is illustrated by experiments with real users in which fewer than ten iterations are usually sufficient to locate an instance from the category of order 100 in a database of size 20,000.

This paper is organized as follows. Related and motivating work is discussed in §2. In §3.1 we formulate interactive search in terms of Bayesian relevance feedback. The answer and display models are introduced in §3.2 and §3.3 respectively. The low-level image descriptors and the clustering algorithm are then described in §4.2 and §4.3. In §4.4, we present the experiments. Finally, we conclude with a summary of our findings and remarks on plausible continuations.

2. Related Work

With the page zero problem in mind, Lesaux *et al.* [14] create a summary of the image database from unsupervised categorization followed by a user-guided refinement of the resulting clusters. Cluster prototypes then provide an overview of the database that can be consulted to find a suitable query point. Fauqueur *et al.* [6] fabricate a query example by composing image patches (regions), utilizing a visual thesaurus composed of many region categories ("sky", "building", "grass", etc.) and logical connectors. However, neither of these works involves relevance feedback.

Li and Wang [12] represent semantic concepts by feature-based probability distributions, allowing for models to be updated as the database grows without massive retraining. Carneiro *et al.* [2] model images as bags of localized feature vectors, estimating a mixture density for each image; the mixtures associated with images with shared annotations are pooled into a density estimate for the corresponding semantic class. Once images are associated with semantic concepts, by whatever method, new queries can be seeded by using natural language or keywords. Even if the annotations are not completely reliable, the user is likely to find a suitable starting point among the retrieved results.

In the area of category search, but assuming a starting point, Caenen and Pauwels [1] assign to each image in the database a probability that reflects its relevance to the user's intentions. The systems is based on a quadratic logistic regression model, used to select the next sample of images that will be presented to the user for individual annotation. There is no mental matching, The shared feature with our work is the image-specific distribution and a statistical framework.

Mental matching seems to have first appeared in the work of Cox *et al.* [3] on iterative search for a specific image in the database (*target search*). We extend the Bayesian framework introduced in that seminal work. At every round, the user is asked to choose which of two images displayed by the search engine is "closest" to the target image residing in his mind. However, the formulation in [3] does not extend to *category search* because the mechanism gathering information ceases to be computationally feasible. Indeed, one cannot maintain a probability distribution on arbitrary *subsets* of images, even for small databases. Also, the answer model does not accommodate more complex user behavior inherent in multiple displays.

Still in the context of target search, and a Bayesian model, Fang and Geman [5] proposed an efficient display algorithm and applied it to mental face retrieval. We shall adapt their display mechanism to our purposes in §3.3. Also, unlike in [3], the answer model is explicitly designed to capture human decision-making (through learned parameters). Still, the approach in [5] does not scale well to large generic (heterogeneous) databases, both computationally and in terms of number of feedback rounds necessary to reach the target. Indeed, the user's notion of similarity is more complex for generic images than for faces, and his choices are less likely to cohere with the feature-based metric employed by the system. Nor does the method in [5] extend to category search.

3. Feedback Framework

Suppose Ω denotes a database of N images, labeled $\{1, 2, \ldots, N\}$ for simplicity. The objective is to identify an image that matches the semantic and visual impressions in the mind of the user. Let $S \subset \Omega$ denote that subset of the database, i.e., the user's category or *target class*. The subset S is unknown to the system. We assume that if a member of S is displayed, the user will recognize it as an instance of the target class, terminating the search. At that point, other members of S could be retrieved by standard query-by-visual-example.

A relevance feedback session is composed of several rounds (iterations) during each of which a set $D \subset \Omega$ of n images is displayed. If $D \cap S \neq \emptyset$, the user identifies an element of his category; otherwise, the user chooses the image in D which he deems to be "closest" to S. Naturally this concept of similarity will only partially cohere with the one employed by the system, which is based on standard color, texture and shape image features (see §3.2).

The most straightforward generalization of the Bayesian framework for target search [3, 5] would be centered on a probability distribution for S and an answer model conditional on S. This distribution would then be updated after each iteration and would drive the display algorithm. Needless to say, this is computationally impossible because, in practice, S is of order 10 to 10^2 and N is of order 10^4. Hence the number of possible subsets is gigantic.

Instead, we associate a binary random variable Y_k to each image $k \in \Omega$: $Y_k = 1$ if $k \in S$ and $Y_k = 0$ if $k \notin S$. Of course, $S = \{k \in \Omega : Y_k = 1\}$, so S and $\{Y_k\}$ carry the same information. We maintain N parallel Bayesian systems, one for each image. Consequently, there is a response model for each k separately, and after each feedback iteration, and for each k, we update the posterior distribution on Y_k given the search history. More specifically, if B_t denotes the responses of the user to the first t displays (see §3.1), then the distribution of Y_k given B_t is represented by the single parameter $p_t(k) = P(Y_k = 1|B_t)$. (We take the starting distributions $p_0(k) = 0.5$ for simplicity.) Notice that $\sum_{k\in\Omega} p_t(k)$ represents $E(|S||B_t)$, the expected size of S after t queries. In particular, p_t is *not* a distribution over Ω.

Our framework has three key components:

- *Update Model*: Computes $p_{t+1}(k)$ in terms of $p_t(k)$ and the user's answer at step $t+1$;
- *Answer Model*: Specifies the probability the user chooses image $x \in D$ given $Y_k = 1$ and given $Y_k = 0$ for each k;
- *Display Model*: Determines which images to display at step t based on $\{p_t(k)\}$ and the search history.

3.1. Update Model

Let X_D denote the user's response to display D, a random variable (see §3.2). The feedback up to iteration t is then

$$B_t = \bigcap_{s=1}^{t} \{X_{D_s} = x_s\} \tag{1}$$

where D_s is the display at step s and x_s is the user's response.

The basic statistical assumption is that the random variables $X_{D_s}, s = 1, 2, ...,$ are conditionally independent given Y_k for each k. (Note: this is different from assuming conditional independence given S, a more natural assumption but not sufficient for our purposes.) Updating each $p_t(k)$ depends on *both* the "positive" and "negative" response models. These are assumed to be time-independent and denoted by $P(X_D = x|Y_k = 1)$ and $P(X_D = x|Y_k = 0)$ respectively. Since $B_{t+1} = B_t \cap \{X_{D_{t+1}} = x\}$ and since $X_{D_{t+1}}$ is independent of B_t given Y_k, we have

$$\begin{aligned} p_{t+1}(k) &= P(Y_k = 1|B_{t+1}) & (2) \\ &= P(X_{D_{t+1}} = x|Y_k = 1)p_t(k)/C_{t+1} & (3) \end{aligned}$$

where the normalizing constant C_{t+1} is $P(X_{D_{t+1}} = x|Y_k = 1)p_t(k) + P(X_{D_{t+1}} = x|Y_k = 0)(1 - p_t(k))$.

3.2. Answer Model

Let $D_t = \{x_1, \ldots, x_n\} \subset \Omega$ be the set of images displayed at iteration t. We can suppress t since the response model is time-invariant. Moreover, we can assume that no element of S appears in D since otherwise the search terminates. Consequently, the response X_D assumes values in D itself: $X_D = x_i$ signifies that image x_i is the closest image to S *in the opinion of the user*.

Let d denote the metric in the signature space. We adopt answer models of the form:

$$P(X_D = x_i|Y_k = 1) = \frac{\phi_+(d(x_i, k))}{\sum\limits_{x_j \in D} \phi_+(d(x_j, k))} \tag{4}$$

$$P(X_D = x_i|Y_k = 0) = \frac{\phi_-(d(x_i, k))}{\sum\limits_{x_j \in D} \phi_-(d(x_j, k))} \tag{5}$$

The design of the functions ϕ_+ and ϕ_- is motivated by the intuitive expectation that the perceived similarity between two images will be roughly inversely proportional to their distance apart in the metric d. Of course the situation is very complex as it involves human decision-making and the efficiency of the model will depend on the extent to which the system metric captures semantic similarity. We take $\phi_+(d)$ (the positive model) to be monotonically decreasing in d and $\phi_-(d)$ (the negative model) to be monotonically increasing in d. As a result, if $k \in S$, the closer the image $x_i \in D$ is to k in the stored metric, the more likely the user is to choose it in the positive model; that is, if $x_i, x_j \in D$ and $d(x_i, k) < d(x_j, k)$ then we expect

$P(X_D = x_i | Y_k = 1) > P(X_D = x_j | Y_k = 1)$. Similarly for the negative model with the inequality on probabilities reversed since we are assuming $k \notin S$.

In our experiments, we adopt parametric forms for ϕ_+ and ϕ_- (see Fig. 1) and learn the parameters from real data (collected user responses) by maximum likelihood estimation.

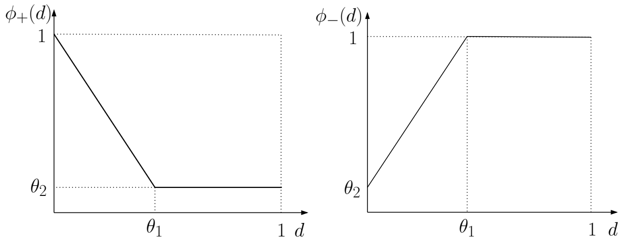


Figure 1. Parametric forms for ϕ_+ and ϕ_-.

The parameter θ_1 can be viewed as a "saturation" threshold: for the positive model (resp., negative model), an image θ_1 units away from a target is no more likely (resp., less likely) to be chosen than one still farther away. The parameter θ_2 controls the degree of coherence between the subjective decisions and the system metric. Take for example the positive model and suppose one displayed image x_i is very close to k and all the other $n-1$ images are farther than θ_1 units from k; i.e., there is one overwhelmingly best choice in terms of d. Then, according to Eq. 4, $P(X_D = x_i | Y_k = 1) \cong 1/(1+(n-1)\theta_2)$. Small values of θ_2 would then imbed high coherence, perhaps unrealistically. We shall return to this issue in §4.4.

3.3. Display Model

Perhaps the simplest procedure for choosing D_{t+1} would be to select the n images most likely to belong to S, as measured by their masses under $p_t(k)$. Unfortunately, this elementary strategy is far less effective (in terms of mean search time) than others due to the fact that it does not take into account visual similarity; for instance, two very similar images, both with high masses, are probably either both in S or both not in S. Put differently, the resulting display does not adequately "sample" the database. Instead, we borrow the line of reasoning in [5], but adapted to category search, and seek a more powerful strategy. We attempt to minimize the uncertainty about S given the search history and the new evidence provided by $X_{D_{t+1}}$:

$$D_{t+1} = \arg\min_{D \subset \Omega} H(S | B_t, X_D) \qquad (6)$$

This combinatorial optimization problem is evidently intractable because it involves looping over all subsets of Ω. But an equivalent reformulation leads to a practical algorithm.

3.3.1 Heuristic Solution

Using elementary properties of conditional entropy,

$$D_{t+1} = \arg\min_{D \subset \Omega} \left(H(X_D | S, B_t) - H(X_D | B_t) \right) \qquad (7)$$

Now imagine an "ideal user" who always chooses the image $x \in D$ which is closest to S in the system metric. That is, this user chooses the image $x_i \in D$ such that $d(x_i, S) \leq d(x_j, S)$ for all $x_j \in D, i \neq j$; here $d(x, S) = \frac{1}{|S|} \sum_{j \in S} d(x, j)$, the average distance to S. In particular, the response X_D of this ideal user is a function of S, and hence $H(X_D | B_t, S) = 0$. As a result, for this user, the optimal display is the one for which $H(X_D | B_t)$ is maximized. Since entropy is maximized at the uniform distribution, there is a natural, sequential procedure for constructing a display D which yields approximately equally-likely answers. The basic idea is as follows. Still assuming our ideal user, we seek n images, call them again $\{x_1, ..., x_n\}$, such that $P(X_D = x_i | B_t) \approx \frac{1}{n}$. Equivalently, we want the Voronoi partition based on these points and on the metric d to have cells of almost equal mass under an appropriate distribution over Ω (see Fig. 2). Since this distribution is inaccessible (because it involves the posterior over subsets S), we replace it by the distribution over Ω whose masses are proportional to $p_t(k)$ and then we use the algorithm described in [5].

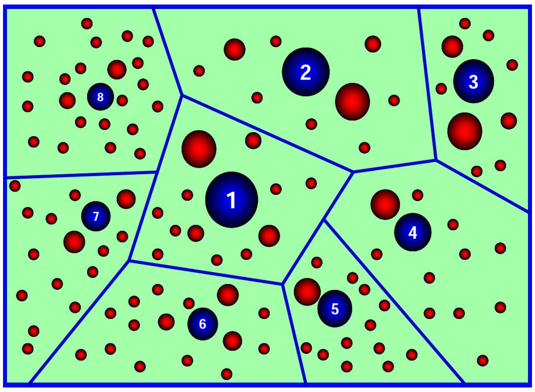


Figure 2. Voronoi partition into 8 cells of equal mass. The size of the images is proportional to their mass. Prototypes are numbered.

3.3.2 Acceleration by Clustering

Although fast and easy to implement, and highly effective for target search, the heuristic solution lacks efficiency for category search with large databases in which many images are visually very similar. In fact, many semantic categories can be very roughly decomposed into a union of clusters of highly similar images. For example, "red flowers" likely have very similar low-level descriptors. Applying the heuristic described in §3.3.1 at the image level can then result in search sessions in which the probability mass gets highly concentrated on images in the complement of S at the beginning of the search session.

For this reason, and in order to keep a distance matrix in memory, we reduce this redundancy by unsupervised clustering of the image database into small but highly coherent cells. Let $\mathcal{C} = \{C_i\}_{i=1}^{p}$ be a partition of Ω. For each cluster $C \in \mathcal{C}$ we compute the expected size of $C \cap S$ given the session history, namely $\eta_t(C) = \sum_{k \in C} p_t(k)$, and then normalize these to a probability distribution $p_t(C)$ over $\mathcal{C}$. We then compute the next display screen D_{t+1} just as previously described, *but at the cluster level*, i.e. feeding the algorithm with the list of clusters $\mathcal{C}$ and the corresponding probabilities $\{p_t(C) | C \in \mathcal{C}\}$ (in place of the $p_t(k)$). The distance between two clusters is the average link distance:

$$d(C_n, C_m) = \frac{1}{|C_m||C_n|} \sum_{i \in C_m} \sum_{j \in C_n} d(i, j).$$

The output of the algorithm is then a list of clusters $\mathcal{D} \subset \mathcal{C}$. For each element $C \in \mathcal{D}$ we choose the image that has the highest posterior $p_t(k)$ for $k \in C$ to be displayed.

Moreover, after the user has chosen an image in $x \in D$, we display the whole cluster containing x, providing the user an opportunity to check the cluster for elements of S. If an element from S is identified, the search session ends. Otherwise, this cluster is eliminated from further consideration; equivalently, $p_{t+1}(k) = 0$ for $k \in C$. The interface is still quite simple: each feedback iteration consists of two steps: (I) the system presents a list D of candidates and the user clicks on the one thought to be most similar to his category; (II) the system presents the cluster associated with the user's choice and the user inspects the cluster for a member of S. Neither step burdens the user, and average search time is somewhat lowered by inclusion of step (II).

4. Experiments

4.1. Image Database and Ground Truth

To test our search engine we use a subset of 20,000 images from the Corel database covering a broad range of semantic themes: agriculture, architecture, cities, closeups, cuisine, landscapes, museum, space, sports, textures, etc. For groundtruth, we selected by hand ten semantically coherent image classes of medium size (100 images per class), ensuring that the interpretation is unambiguous; see Fig. 3.

4.2. Image Descriptors

To describe the low-level visual content of the images, we employ the weighted histograms described in [15], using the Laplacian and the local probability of colour as pixel weighting functions. Weighting functions bring additional information into the histograms (e.g. local shape or texture), which is an important principle in building reliable image descriptors. The resulting integrated image descriptors generally perform better than a combination of classical, single-aspect features. Moreover, weighted histograms work equally well for color images and for gray level images.

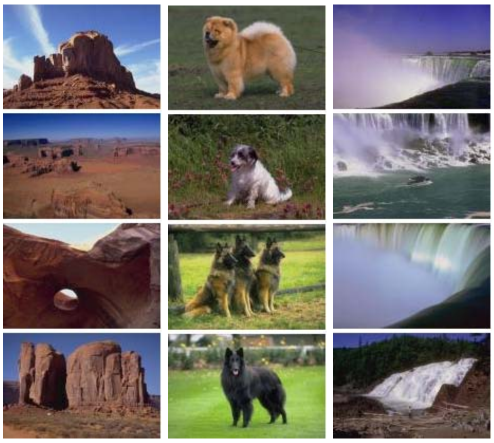


Figure 3. Samples from three semantic groundtruth classes: “Monument Valley” (left), “pedigree dogs” (middle), ‘waterfalls” (right). The other groundtruth classes are: “African antelope”, “butterfly”, “doors of Paris”, “fireworks”, “deep forest”, “molecule” and “owl”.

To describe the shape content of an image we use a histogram based on the Hough transform, which captures the behavior along straight lines of varying directions and performs better that the classic edge orientation histogram [10]. Texture feature vectors are based on the Fourier transform; we use the distribution of the spectral power density along the frequency axes [13]. Finally, the metric in the combined feature space is the L1 distance, normalized to values in $[0, 1]$; results with L2 are similar, but slower.

4.3. Clustering

Recall that our display model is based on clustering the database. Since the database is generic, and since no prior information about semantic content is available, smaller clusters are expected to be more coherent than larger ones. Needless to say, the elements of even a small cluster may belong to different semantic classes. However, this is not a problem since we maintain a list of probabilities $p_t(k)$ at the image level. Since whole clusters are presented to the user after responding to a display screen, the clusters should be small enough to be rapidly inspected, even if several clusters are visually similar.

We tried several classical clustering algorithms, such as K-Means, Fuzzy K-Means [4] and Competitive Agglomeration [7]. However, the results were inadequate for our purposes because some quite large clusters (more than 100 images) were generated with highly diverse visual and semantic structure.

To satisfy our requirements, we developed a modification of Quality Theshold clustering [9], which provides control over the size of the clusters and is independent of initialization. Briefly, given a desired cluster size K, the algorithm iteratively chooses new clusters from a list of candidates based on computing the K nearest neighbors to each unclustered image. The candidate with the smallest diameter (englobing sphere) is chosen. Running time is no issue since the computation is offline. In Fig. 4 we show some example clusters of size eight. Most clusters are visually consistent in terms of our signatures based on color, texture and shape. Semantic diversity is tolerated since we only use the clusters to simplify the display algorithm. Of course the more homogeneous semantically the better, the ideal being that every semantic category be a perfect union of clusters. This however can never be guaranteed.

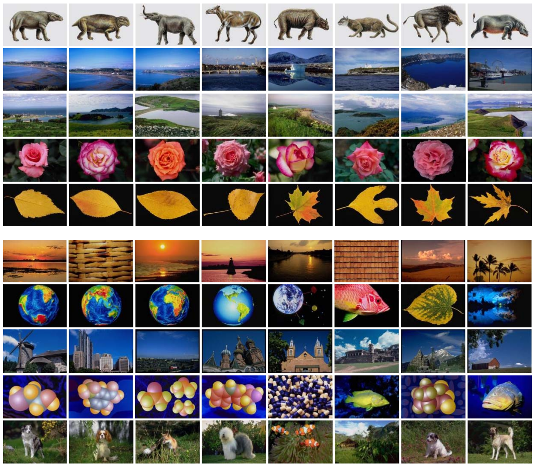


Figure 4. Some clusters of size 8: some are very coherent semantically (top rows) whereas others are less so (bottom rows).

4.4. Experiments with Real Users

We tested our method for interactive search by collecting responses from a group of 12 individuals not familiar with the system. For each individual, and each of the ten groundtruth classes, the user is presented with a visual summary of his target class and a new relevance feedback session is started with a random display. The session ends when an element of the target class is identified. Every (non-terminal) click provides a "data item" in the sense of a triple (S, D, x) corresponding to a target class, set of displayed images and user's response. We set $n = |D| = 8$; displaying many fewer or many more images has adverse consequences with real users.

Parameter estimation. We collected 652 data items, (S_i, D_i, x_i), from 12 users and estimated the parameters θ_1 and θ_2 appearing in ϕ_+ and ϕ_- (see §3.2) by maximum likelihood. For $\phi_+(d; \theta_1, \theta_2)$ we maximize the likelihood function:

$$L_+(\theta_1, \theta_2) = \prod_i P_i(S_i, D_i, x_i)$$

where P_i is the probability of response x_i to display D_i given the target class is S_i:

$$P_i(S_i, D_i, x_i) = \frac{\phi_+(d(x_i, S_i))}{\sum_{x_j \in D_i} \phi_+(d(x_j, S_i))}$$

Notice that we have used a simplified version of the stochastic answer model in Eq. 4 which utilizes the entire target class S (but is not the deterministic "ideal user" from §3.3.1). For the negative model $\phi_-(d; \theta_1, \theta_2)$ we maximize a similar likelihood function, namely $L_-(\theta_1, \theta_2) = \prod_i P_i(S_i, D_i, x_i)$, where:

$$P_i(S_i, D_i, x_i) = \frac{\phi_-(d(x_i, \Omega \setminus S_i))}{\sum_{x_j \in D_i} \phi_-(d(x_j, \Omega \setminus S_i))}$$

ϕ	θ_1	θ_2
ϕ_+	0.40	0.06
ϕ_-	0.26	0.29

Table 1. Optimum values for the parameters θ_1 and θ_2

For our set of users and our database, we obtained the parameter estimates presented in Table 1. In particular, the estimated values for the positive model suggest that "distance saturation" occurs at (relative) distance $\theta_1 = 0.4$ in the description space. For the situation of a unique, near-perfect match (see §3.2) the estimated probability of selecting the good match in signature space is $1/(1 + 7 \cdot 0.06) = 0.70$, which, despite being based on real data, appears to us to be an overly optimistic measure of the coherence between a real user and the system metric. However, such displays are quite rare in real search sessions (see the following experiment).

Coherence analysis. To assess the influence of the size of the clusters we performed experiments with sizes 8 and 20. Using larger clusters makes for an unpleasant display and burdens the user. We collected data from a set of 12 users: 790 items (clicks) for size 8 and 667 items for size 20. From this, we analyzed the coherence between the user's notion of similarity and the employed metric. Recall that the cluster size influences the display algorithm as the metric is computed at the cluster level. Fig. 5 depicts the probability that the user selects the image which ranked first, second, etc. relative to the system metric, where the displayed images are ranked according to their distance to the target class. As we can see, the metric induced by the image descriptors is not highly coherent with mental matching; for example, the probability the user selects the closest image to the target class is only roughly 0.22. Nevertheless, the departure from the uniform distribution is sufficiently large to

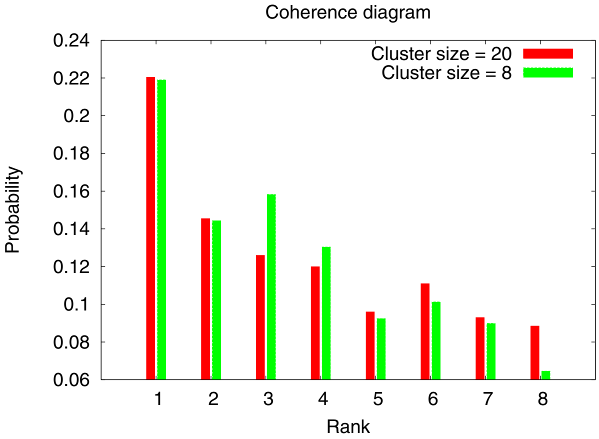
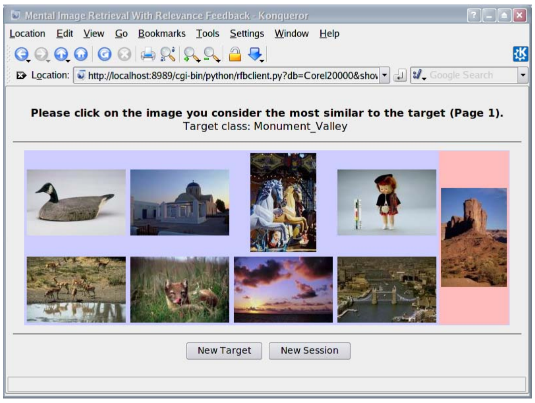


Figure 5. The probability that the user selects the m-th closest image to the target.

convey enough information to yield very reasonable search times.

Performance. The experimental interface is shown in Fig. 6. The key quantity is T, the number of iterations during a search (relevance feedback) session required to locate an instance from the target class. We estimate $E(T)$, the mean of T, and $P(T \leq t)$, the (cumulative) distribution of T, by their empirical statistics collected over M search sessions. The cluster size is $n = 8$. Evidently, the faster $P(T \leq t)$ grows, the more efficient the system is operating.


Mental Image Retrieval With Relevance Feedback - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: http://localhost:8989/cgi-bin/python/rfbclient.py?db=Corel20000&show
Google Search

Please click on the image you consider the most similar to the target (Page 1).
Target class: Monument_Valley

Figure 6. The interface used for experiments.

As benchmarks, we also present the results of two simulations under the same experimental settings (same groundtruth classes, etc.) of two extreme cases: the "ideal user" and the "random user." Recall that the ideal user always chooses the image closest to the target class in the system metric. This represents the optimal performance we can hope to attain. The other extreme is a random response; the user selects one of the eight displayed images at random. Obviously, the proposed model far out-performs a random response. More importantly, the absolute performance is quite reasonable (Fig. 7), with a mean search time $E(T) \approx 8$ and target recovery in fewer than four iterations in approximately one-half the searches and in fewer than ten iterations in more than seventy percent of the searches. Fine-tuning the model might result in still better performance. The results for cluster size $n = 20$ are similar: $E(T) = 5.7, 6.75, 21.57$ for the ideal user, real user and random user, respectively.

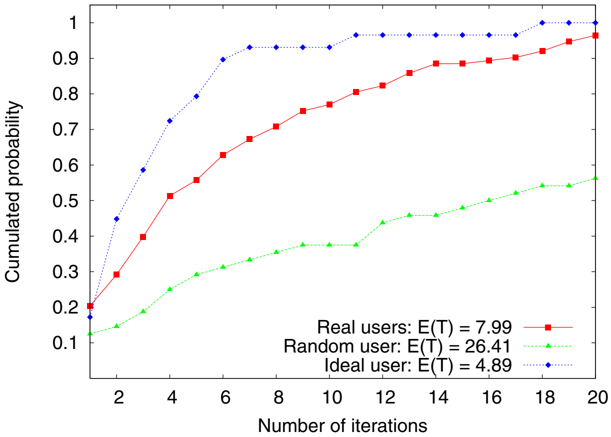


Figure 7. Cumulative distribution of the search time for real, ideal and random users.

Returning to the page zero problem, the baseline is the random display of images, without replacement, until a member of the target class appears. Computing the average number of screens necessary is then a relatively straightforward exercise in probability. If N, L, n are the sizes of the database, target class and display, respectively, then the mean $E(T) \cong N/n(L+1)$. In our experiments, $N = 20,000, L = 100, n = 8$; hence the average is around 25 iterations. Accounting for the display of the cluster containing the user's selection would lower this average, but not nearly by half since the clusters are so visually coherent, which works against rapid discovery. Indeed, a displayed cluster is not a random subset from the database, and is not independent from the preceding display. In fact, for a cluster size of 8 (Fig 7), the performance of the random user, namely $E(T) = 26.41$), which includes the cluster display, does not improve at all relative to the baseline mean of approximately 25 given above due to the coherence of the clusters.

To test our algorithm on a professional, unstructured database, we performed experiments on a database of 19500 art images, kindly provided by Alinari (http://www.alinari.it). Although this database is more complex than Corel (see Fig. 8), we obtain similar results to those in Fig. 7, but with a smaller population of users and search sessions.

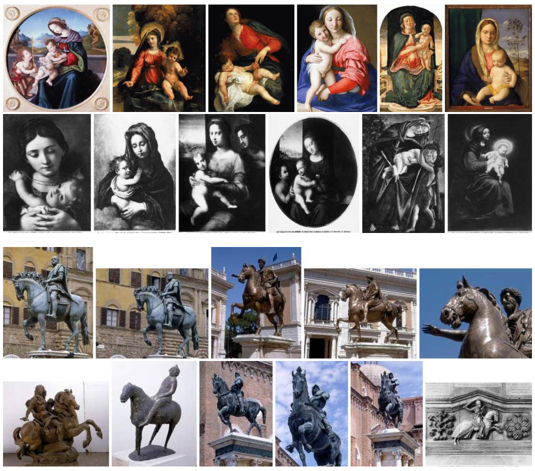


Figure 8. Samples from two classes (Alinari database): “Madonna and Child” (top rows), “Horse and Rider” (bottom rows).

5. Conclusion

We have presented a Bayesian framework for discovering an instance of a semantic category residing in a large, unstructured database using relevance feedback. Since the category is known only to the user of the system, the feedback is based on mental matching. Our framework centers on an evolving estimate of the probability that each member of the database belongs to the user’s category. The update mechanism take advantage of the conditional independence assumptions on the sequence of responses provided by the user. A central feature is the new Bayesian model, which includes a pair of positive and negative answer models which are designed to account for subjectivity of the user’s choices and their weak correlation with the system metric. The performance of the system is validated on a database of 20,000 images; experiments with real users demonstrate the feasibility of the proposed model.

Both the answer and display models could likely be improved. In the former case, two parameters may not be enough to capture the first-order effects in human decision-making and variability among users. As for the display, one issue we encountered during our tests was the difficulty users have in deciding which image to select when *all* the displayed images appear “semantically distant” from the target category. In this case, the user’s choice is likely to be highly random relative to the system metric. Higher search time efficiency might then be achieved by allowing the user to reject the whole screen when it is felt that there is no natural choice. We are currently investigating this and other extensions.

References

[1] G. Caenen and E. J. Pauwels. Logistic regression model for relevance feedback in content-based image retrieval. In *Proc. Storage and Retrieval for Media Databases*, pages 49–58, 2001.

[2] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):394–410, 2006.

[3] I. J. Cox, M. L. Miller, T. P. Minka, T. Papathomas, and P. N. Yianilos. The Bayesian image retrieval system, PicHunter: theory, implementation and psychophysical experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, January 2000.

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd ed.)*. Wiley Interscience, 2001.

[5] Y. Fang and D. Geman. Experiments in mental face retrieval. In *Proc. of Audio- and Video-based Biometric Person Authentication*, pages 637–646. Lecture Notes in Computer Science, 2005.

[6] J. Fauqueur and N. Boujemaa. Mental image search by boolean composition of region categories. *Multimedia Tools and Applications*, 31(1):95–117, 2006.

[7] H. Frigui and R. Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):450–465, 1999.

[8] T. Gevers and A. W. M. Smeulders. Content-based image retrieval: An overview. In G. Medioni and S. B. Kang, editors, *Emerging Topics in Computer Vision*. Prentice Hall, 2004.

[9] L. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome*, 9(11):1106–1115, 1999.

[10] A. Jain and A. Vailaya. Shape-based retrieval: a case study with trademark image databases. *Pattern Recognition*, 31(9):1369–1390, 1998.

[11] M. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State-of-the-art and challenges. *ACM Transactions on Multimedia Computing, Communication, and Applications*, 2(1):1–19, 2006.

[12] J. Li and J. Z. Wang. Real-time computerized annotation of pictures. In *Proceedings of the ACM Multimedia Conference*, pages 911–920, 2006.

[13] B. Manjunath, P. Salembier, and T. Sikora, editors. *Introduction to MPEG-7: Multimedia Content Description Interface*. Wiley, 2002.

[14] B. L. Saux and N. Boujemaa. Image database clustering with SVM-based class personalization. In *SPIE Conference on Storage and Retrieval Methods and Applications for Multimedia, part of Electronic Imaging symposium*, 2004.

[15] C. Vertan and N. Boujemaa. Upgrading color distributions for image retrieval: can we do better? In *International Conference on Visual Information Systems (Visual2000)*, November 2000.

[16] X. S. Zhou and T. S. Huang. Relevance feedback for image retrieval: a comprehensive review. *Multimedia Systems*, 8(6):536–544, 2003.