A Stochastic Feedback Model for Image Retrieval

D. Geman[1] R. Moquet[2]

[1] University of Massachusetts and Ecole Polytechnique
[2] Ecole Polytechnique

Department of Mathematics and Statistics
University of Massachusetts
Amherst, MA 01003
geman@math.umass.edu

Abstract

Our goal is an efficient algorithm for image retrieval based on relevance feedback. We assume the user is searching for a particular image in a database and responds to a sequence of machine queries by declaring which of two (or more) displayed images is "closest" to his target. Efficiency is measured by the average number of queries necessary to locate the image. We introduce a Bayesian feedback model which accounts for considerable variation in the responses of the user through a sequence of independent random metrics on feature space whose distribution may depend on both the displayed images and the target. Each new query is chosen to minimize the expected conditional entropy of the distribution over targets given the previous responses. The resulting algorithm is demonstrated for shape and image retrieval and its performance compared with theoretical bounds and previous models.

Keywords

image retrieval, relevance feedback, decision tree, random metric, Bayesian model, entropy reduction.

1 Introduction

Recently, the number of images stored numerically, and the number of people searching for particular ones in large databases, have grown significantly. One problem is to design an interactive procedure by which a "user" can retrieve a desired image(s) with a reasonable amount of effort (and before giving up). Unfortunately, natural global descriptions, such as *"a landscape with a river behind a small cottage,"* are virtually impossible to match to specific images in the database since the problem of automatically extracting semantic descriptions of natural scenes is largely unsolved. Consequently, systems for image retrieval are often based on a sequence of queries put to the user in order to garner information about attributes of the desired image(s). Rarely does the user have any technical knowledge about images, such as the manner in which they are stored and analyzed. The answers are inevitably subjective and the interaction is inherently stochastic. Indeed, this is the most distinctive aspect of the image retrieval problem.

The problem is reasonably well-solved for written documents, such as books in libraries or on web sites. A query might consist of key word(s) supplied by the user. The system then displays the matching documents, either by searching the documents themselves or pre-processed indices. Automatic indexing of text is feasible, and searching can be made efficient, by exploiting the information residing in the statistical distribution of words and other verbal constructs. It is not apparent how to extend this to images. One could index them by "key words" and ask the user to supply appropriate ones from a given list. But manual indexing would likely be necessary and a list of individual words, however well-chosen, carries relatively less information for identifying images than for text.

1.1 Previous Work

An alternative is "content-based indexing." Each image in the database is represented by a set of feature vectors based on color, edge and other image statistics. In this way, one can compute a "distance" $d(y, y')$ between two images y and y' based on standard metrics adapted to the individual features. Ideally, $d(y, y')$ is "small" when y and y' "look alike" to human beings.

Once the features and the distance are determined, a variety of search protocols are possible. For example, the user might be asked to select one image from a displayed list and then the system might display the k nearest neighbors. Or the user might be asked to select the images which are, *in his opinion*, closest to the specific image, or general type, he is seeking. The system then then displays another set of images, hopefully more homogeneous and closer to the target, and the interaction continues until the target image is displayed (and presumably recognized). Most image retrieval algorithms employ some variation of such *relevance feedback*, introduced in [4]; two examples are the Surfimage system based on "category search" [5] and the PicHunter system based on "target search" in [3].

We have adopted the Bayesian framework in [3] based on stochastic comparison search. The goal is to retrieve one target Y, considered to be a random variable with some distribution. In [3], a distance d is fixed and at each step a pair of images y, y' is displayed; the user chooses the one he finds most similar to Y. If his choices were exactly based on the metric d, the response would be y if $d(y,Y) < d(y',Y)$ and y' otherwise. Instead, to account for subjectivity, the authors consider a "blurring" of the true answer and the actual response is modeled as a random variable whose probability distribution depends on $d(y,Y) - d(y',Y)$. The current probability distribution on Y is then updated based on the latest response and a new pair of images is selected based on the updated distribution over targets.

1.2 Our Approach

The real interactive process is more fundamentally random than a noisy response to a fixed, known metric. In our model, the metric is itself a random variable, and unknown. In fact, the responses of the user are based on a *sequence of independent random metrics* in feature space whose distribution may depend both the target Y and the displayed images. The different metrics correspond to different weightings of the individual features. We wish to accommodate the fact that some attributes are typically weighted more heavily than others depending on both what the user has in mind and what he sees. Each new query is chosen to minimize the expected conditional entropy of the distribution over targets given the previous response. This posterior distribution on targets then evolves according to the new response. The resulting algorithm is demonstrated for shape and image retrieval and its performance compared with theoretical bounds, ideal scenarios and previous models. There is a clear gain in efficiency due to generality.

2 Indexing and Comparing Images

Let $\mathcal{Y} = \{y_1, ..., y_n\}$ denote the database of images. One of these, Y, is the image the user has in mind and our goal is to find it as quickly as possible by asking series of questions based on displaying elements of $\mathcal{Y}$. Many types of queries are possible. One could display k images at each step and ask the user to choose the one "closest" to his target. Starting with a random set, the displayed images could be the k nearest neighbors in $\mathcal{Y}$ under some metric on images. Or one could ask the user to select the subset of displayed images which are "most relevant." In a comparison search exactly two images are displayed at each step and the user selects the one which is, in his opinion, closer to his target. This process continues until one of the two displayed images is the target; we assume the user recognizes it and the search is terminated.

From the point of view of the user, there are only images. However, all the computation performed by the system - namely choosing which images to display and updating a probability distribution on $\mathcal{Y}$ - is based on an "index" $f(y)$ assigned to each image $y \in \mathcal{Y}$ and a metric for comparing images based on this index. The images in $\mathcal{Y}$ are automatically preprocessed in order to extract discriminating features grouped into broad classes. The index of y is then of the form $f(y) = (f_1(y), ..., f_m(y))$. We will refer to m as the *dimension* of the index. Each feature $f_s(y)$ is itself an vector of real numbers, say of dimension p_s, computed from the raw intensity data and represents one local or global characteristic of y. Some common examples are color histograms, Fourier or wavelet coefficients, texture attributes and edge statistics (e.g., orientation histograms).

The distance between two images is based on the feature vector. Let d_s be a metric on $\mathbf{R}^{p_s} \times \mathbf{R}^{p_s}$, $s = 1, ..., m$. These metrics are fixed throughout. For each set of positive coefficients $\alpha_1, ..., \alpha_m$, define a metric on images by

$$d(y,y') = \sum_{s=1}^{m} \alpha_s d_s(f_s(y), f_s(y')),$$

and let $\mathcal{D}$ denote the space of all such metrics generated by coefficients for which $\sum_s \alpha_s = 1$. We will assume that the procedure for comparing images em-

ployed by the user is actually *some* element of $\mathcal{D}$, which might change from query to query, and might depend on both the target and the query. In contrast, in [3] there is one frozen metric.

We can now specify the set of queries. For each $d \in \mathcal{D}, 1 \le i < j \le n$ and $y \in \mathcal{D}$, define

$$X_{ij}(d,y) = \begin{cases} 1 & \text{if } d(y_i,y) < d(y_j,y) \\ 0 & \text{otherwise} \end{cases}$$

Here i,j refer to the pair of displayed images and y is thought of as the target.

3 A Statistical Model

We will construct a joint probability distribution for queries and targets. This involves a family of auxiliary random variables based on the metrics. The random variables are:

Target Variable: The target Y is a random variable with marginal (or "prior") distribution $p_0(y), y \in \mathcal{Y}$. This distribution "evolves" as information is collected from queries. In all our experiments we take p_0 to be uniform.

Random Metrics: A family $\{D_{ij}\}$ of random variables with values in $\mathcal{D}$ and indexed by pairs $1 \le i < j \le n$; D_{ij} determines the answer to a query if images y_i and y_j are displayed. The conditional distribution of the family given Y is determined by two properties:

1. The $\{D_{ij}\}$ are conditionally independent given Y.
2. The conditional distribution of D_{ij} given Y is

$$P(D_{ij} = d | Y = y_l) = \mu_{ijl}(d),$$

where $\{\mu_{ijl}\}$ are fixed probability distributions on $\mathcal{D}$.

These two properties imply that

$$P(D_{ij} = d_{ij}, 1 \le i < j \le n | Y = y_l) = \prod_{ij} \mu_{ijl}(d_{ij}).$$

Queries: The queries are now random variables. The complete family is $\{X_{ij}(D_{ij}, Y)\}$, one query X_{ij} for each pair i,j of images.

It is then easy to compute the conditional distribution on any subfamily of queries given Y. Let $D_{ijl} = \{d \in \mathcal{D} : d(y_i,y_l) < d(y_j,y_l)\}$, the subset of metrics for which the answer is "yes" to the query $X_{ij}(d,y_l)$. Then for any $(i_1,j_1),...,(i_k,j_k)$ and sequence of answers $x_1,...,x_k \in \{0,1\}$:

$$P(X_{i_1j_1} = x_1, ..., X_{i_kj_k} = x_k | Y = y_l) = \prod_{r=1}^{k} \mu_{i_rj_rl}\left(D_{i_rj_rl}^{x_r}\right) \quad (1)$$

where $D^1 = D$ and $D^0 = D^c$.

4 Query Selection

The procedure for selecting the two displayed images is recursive, based on the current "posterior" distribution on targets. Suppose k pairs of images, denoted $(i_1,j_1),...,(i_k,j_k)$, have been displayed, resulting in (binary) answers $x_1,...,x_k$. The "testing history" is then

$$B_k = \{X_{i_1j_1} = x_1, ..., X_{i_kj_k} = x_k\}$$

and the posterior distribution is

$$p_k(y) = P(Y = y | B_k), y \in \mathcal{Y},$$

computed via Bayes formula from p_0 and (1). We have experimented with several protocols, such as random sampling and displaying the images y_i, y_j with the two largest masses under p_k.

The consistently most efficient procedure, albeit intensive, is stepwise entropy reduction, the standard recipe for constructing decision trees in machine learning and statistics [1]. At step $k = 1$,

$$(i_1,j_1) = \arg\min_{ij} H(Y | X_{ij}(D_{ij}, Y))$$

where $P(D_{ij} = d, Y = y_l) = \mu_{ijl}(d)p_0(y_l)$. (Here, $H(U|V)$ denotes the conditional Shannon entropy [2] of U given V, i.e., the expectation with respect to p_V of $H(U|V = v)$.) The first pair of displayed images is then y_{i_1}, y_{j_1} and we model the user's response x_1 as the answer to the query $X_{i_1j_1}(D_{i_1j_1}, Y)$. The $k+1$'st query *depends on the previous ones* and is determined by

$$(i_{k+1}, j_{k+1}) = \arg\min_{ij} H(Y | B_k, X_{ij}(D_{ij}, Y)). \quad (2)$$

For each fixed ij, the entropy in (2) is determined by the distribution of $(Y, D_{i_1j_1}, ..., D_{i_kj_k}, D_{ij})$.

In the following three sections we consider three examples in more detail:

- **One Fixed Metric:** The "baseline" case (least amount of uncertainty). The distributions μ_{ijl} are all point masses.
- **One Fixed Distribution:** The distributions μ_{ijl} are all identical but non-degenerate.
- **The General Case:** The distributions are display-dependent and target-dependent.

We will argue, and attempt to demonstrate, that the general case is in fact the most realistic (as a model for human subjectivity) and in fact the most efficient in experiments with people.

5 One Fixed Metric

This is the "ideal" case - one fixed metric $d^* \in \mathcal{D}$ which determines every user response. In the notation above, $\mu_{ijl} = \delta_{d^*}$ for all ijl and hence $D_{ij} \equiv d^*$. In addition, this metric is known (by the system). (One could also imagine trying to estimate it during a training phase, as in [3].) Therefore, the only source of randomness is Y, which then determines the responses $x_1, ..., x_k$, and for any testing history B_k, either $P(B_k|Y = y) = 1$, which occurs if $x_1, ..., x_k$ are consistent with using d^* in the queries $X_{ij}(d^*, Y)$ appearing in B_k, and $P(B_k|Y = y) = 0$ otherwise.

The conditional entropy simplifies to

$$H(Y|B_k, X_{ij}(d^*, Y)) = H(Y|B_k) - H(X_{ij}(d^*, Y)|B_k).$$

Since $H(Y|B_k)$ is independent of ij, minimizing the lefthand side is equivalent to maximizing the last term on the righthand side. Further, since the posterior distribution $p_k(y) = P(Y = y|B_k)$ is uniform on the subset of targets which are consistent B_k, and since X_{ij} is binary, this maximization is equivalent to choosing the query which divides the active targets (i.e., those with positive mass under p_k) as evenly as possible.

Let E_n denote the expected number of queries until Y is uniquely determined. It can be shown [6] that

$$E_n = \log_2 n - 2 + O(\frac{\log_2 n}{n}), \quad n \to \infty. \qquad (3)$$

Of course this is the result expected from coding theory since $H(Y) = H(p_0) = \log_2 n$. (The universal bound $E_n \geq H(p_0)$ [2] is violated due simply to the supplementary feedback provided when the user terminates the search.)

This strategy was executed in [6] on artificial databases (points in Euclidean space) of various sizes n and dimensions m, and with various priors p_0. Extensive experiments show that the only meaningful parameter is n, the size of the database. In particular, the dependence on m is minimal. One simulation is given in Figure 1; the results match (3) quite well.

Needless to say, postulating any given metric d^* as the source of the answers to real queries is not realistic. Nor is it realistic to compel the user to "explain" his choices, and thereby try to infer an approximating

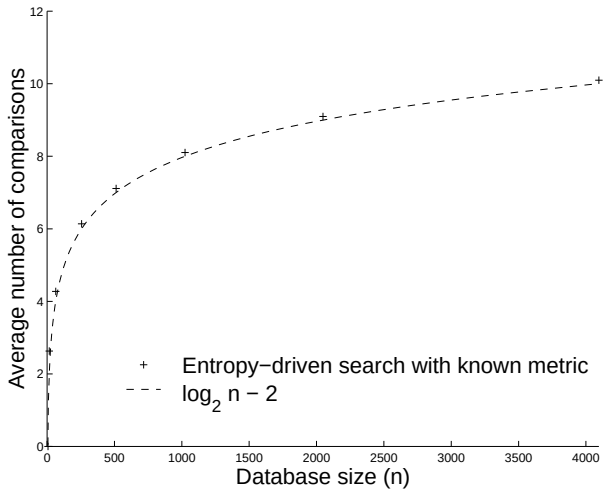

FIG. 1 – *Dependence of search depth on database size in the ideal case.*

metric. Even assuming d is *random*, but stays fixed, i.e., that d is chosen from a fixed distribution on $\mathcal{D}$ but remains the same for all queries, does not lead to good performance in experiments with people, *although better than with d fixed and non-random*; see the results in [6]. In the next section we move one step closer to capturing the level of uncertainty in actual interactions.

6 One Fixed Distribution

In this section we assume the metrics all have the same distribution: $\mu_{ijl} \equiv \mu$. In other words, the random variables $\{D_{ij}\}$ are independent and identically distributed, and independent of Y. In this case, it follows from (1) that the posterior distribution is

$$p_k(y_l) = \frac{p_0(y_l) \prod_{r=1}^{k} \mu\left(D_{i_r j_r l}^{x_r}\right)}{\sum_{l=1}^{n} p_0(y_l) \prod_{r=1}^{k} \mu\left(D_{i_r j_r l}^{x_r}\right)}. \qquad (4)$$

If p_0 is uniform on $\mathcal{Y}$ and μ is uniform on $\mathcal{D}$, which we assume in our experiments, we have

$$p_k(y_l) \propto \prod_{r=1}^{k} |D_{i_r j_r l}^{x_r}|. \qquad (5)$$

We estimate the posterior distribution and the entropy values by Monte Carlo sampling. For each $y_l \in \mathcal{Y}$, we randomly sample a fixed number of metrics in $\mathcal{D}$ and count the number C of these which satsify the inequality appearing in the definition of D_{ijl}; for sufficiently large C this gives a reasonable approximation to the

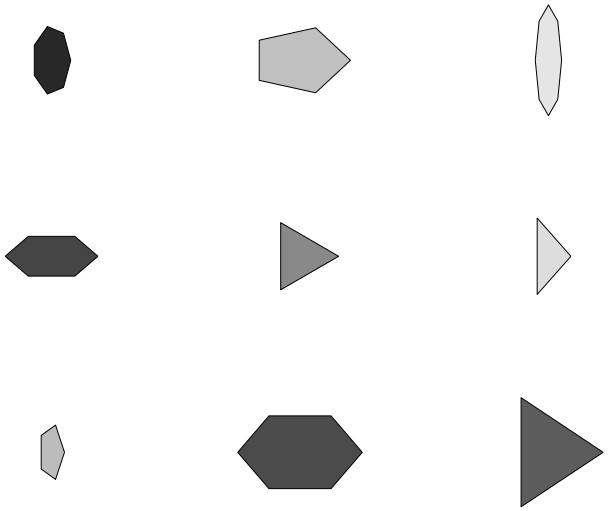

FIG. 2 – *A sample of polygon images.*

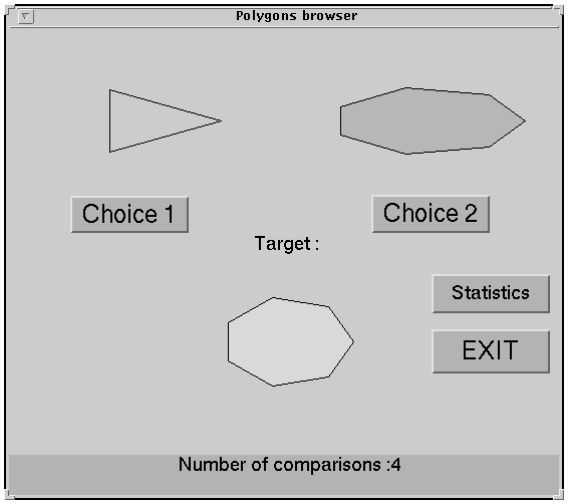

FIG. 3 – *The user interface for finding a polygon.*

volume in (5). The posterior distribution is then easily obtained by normalization. Estimating the entropy is more intensive because it involves estimating p_k with $X_{ij} = 0$ or $X_{ij} = 1$ adjoined to B_k for each possible pair (i, j). Instead of examining all (i, j) in (2), we take a random sample of pairs for which y_i, y_j which have the largest masses under p_k and then select the pair which minimizes the entropy. The results are nearly the same as with a full search; some additional details may be found in [6].

We have performed three types of experiments. The first type, mentioned in the preceding section, is entirely artificial in that the elements of $\mathcal{Y}$ *are* the feature vectors and the responses are generated automatically (i.e., without human intervention) from the model. The second type represents an intermediate step between the first type (simulated objects and queries) and the third type, namely people searching for images. We replace full, complex scenes by a single geometric objects - polygons. In this way we can do controlled experiments to compare the performance (measured by search length) of the various levels of model generality.

Images contain one polygon characterized by four features: size, number of vertices, darkness and "flatness." Thus there are $m = 4$ scalar features. Although simple, such images are sufficiently "meaningful" to allow humans to make comparisons among them. In Figure 2 we show a sample of randomly generated polygons.

To assess the performance of the fixed distribution model, a database of polygons was randomly and uniformly generated. By way of a user interface (see Figure 3), a randomly chosen target is displayed, and the user is asked to answer a series of questions of the form "Which of these two polygons is closer to the target?". (Of course the user is not told that the polygons are represented by the four features.)

It is instructive to compare real and simulated queries, that is, human responses and computer-generated responses in accordance with the model. Evidently, there is less uncertainty when the metric is fixed and random; therefore it is not surprising that, in the case of *with simulated answers*, the search length increases when a new metric is chosen at each query. In Figure 4 we compare the search length for various database sizes and several protocols with simulated answers. In Figure 5 we compare the same protocols with real answers (polygon world). Notice that, in contrast with Figure 4, varying D now performs considerably better than $D_{ij} \equiv D$. Apparently, accommodating more variation in the responses more than offsets the additional complexity.

Finally, since computing or estimating entropies is computationally intensive, we tried a much simpler method of query selection - sampling two images from p_k. As indicated earlier, the posterior distribution is very easy to estimate and sampling from it is virtually instantaneous. Unfortunately, entropy reduction performs appreciably better. One way to characterize the difference is in the evolution of the posterior distribution. The lefthand panel of Figure 6 displays, from top to bottom, the histograms of p_k for $n = 100$ polygon

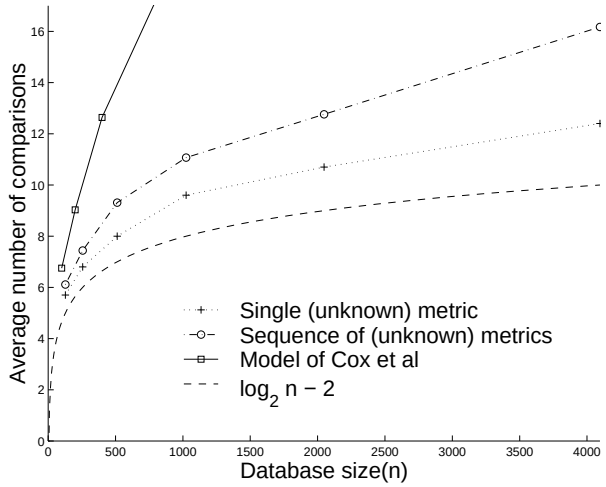

FIG. 4 – *Comparison of models for simulated queries.*

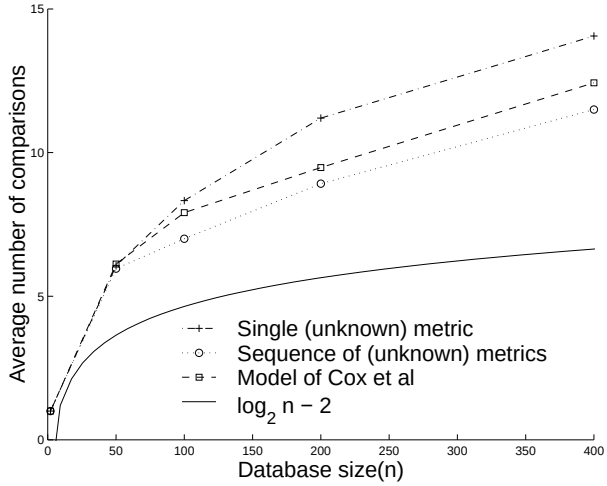

FIG. 5 – *Comparison of models for real queries.*

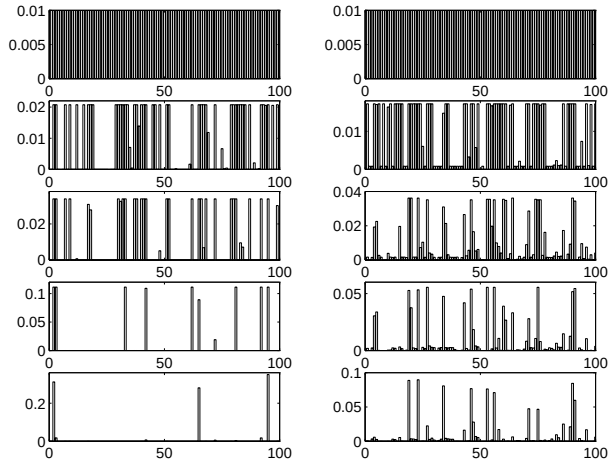

FIG. 6 – *Evolution of the posterior distribution when queries are selected by entropy (left) minimization and random sampling (right).*

images after $k = 0, 1, 2, 3, 4$ queries; query selection is based on entropy reduction. The righthand panel is the same thing with random sampling. Clearly p_k "peaks" more rapidly with entropy.

7 Dedicated Distributions

Until this point the probability distributions of the metrics have been independent of both the target and the displayed images. Surely this is unrealistic in human interactions. If a person has a green disk in mind and is shown a red disk and a blue square, he will likely base his answer on shape and declare the red disk to be closer to his target; whereas if he is shown a green triangle and a blue square he will likely choose the green triangle due to color. And similarly with attributes of real images.

We have performed experiments in the polygon world with sequences of non-identically distributed metrics where the distribution μ_{ijl} depends on an interaction between the target and the displayed images. For polygons, the possible metrics are

$$d(y, y') = \sum_{s=1}^{4} \alpha_s d_s(f_s(y), f_s(y')),$$

where $\sum_s \alpha_s = 1$ and $d_1, ..., d_4$ are simply Euclidean distances on the quantized range of values of the four attributes, normalized to have $\max_{y,y'} d_s(f_s(y), f_s(y')) = 1$. The probability distribution of $(\alpha_1, ..., \alpha_4)$ is based on $d_s(y_i, y_l)$ and $d_s(y_j, y_l)$. Given the display is y_i, y_j and the target is y_l, we assign α_s the uniform distri-

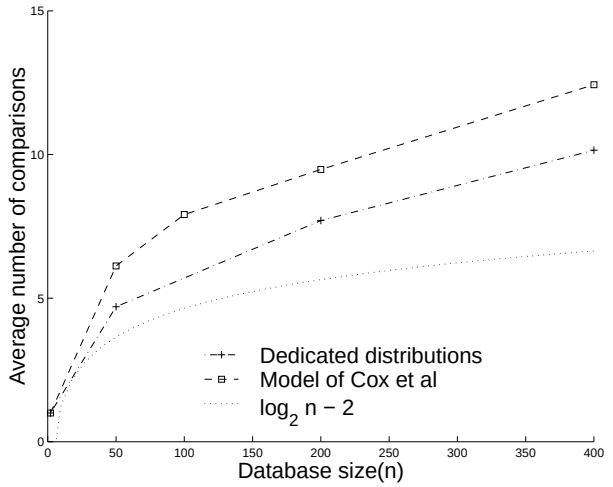

FIG. 7 – *Allowing the probability distribution of answers to depend on the displayed images.*

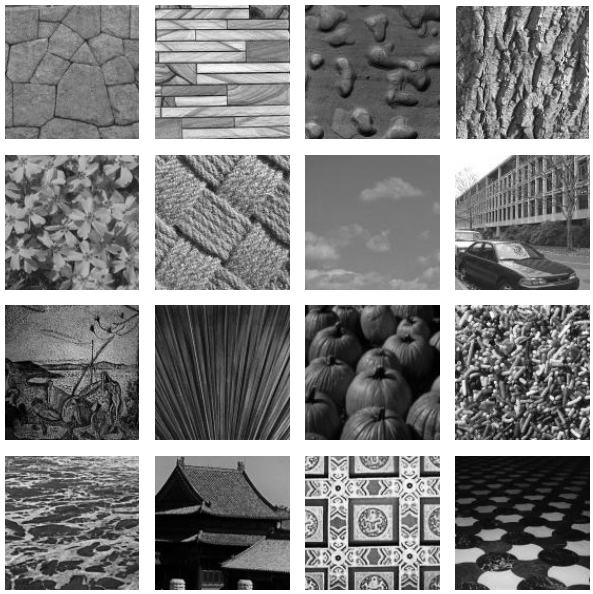

FIG. 8 – *A sample of images from the database*

bution on the interval

$$[0, |d_s(f_s(y_i), f_s(y_l)) - d_s(f_s(y_j), f_s(y_l)|].$$

In this way, attributes are emphasized for which the target is much closer to one displayed image than the other. One experimental result is given in Figure 7. There is a pronounced improvement in the mean search length compared with Figure 6.

8 Experiments with Real Images

To date, we have only done a few experiments with one small image database. It was provided by the IMEDIA project at INRIA-Rocquencourt and consists of 166 pre-processed images clustered into approximately 20 subjects. A sample of these images appears in Figure 8. There are three features for each image with individual dimensions as real vectors ranging from 64 to 90.

Experiments are based on the second model, i.e., a sequence of independent and identically distributed metrics. In particular, the metric distribution depends neither on the target nor the two displayed images. Implementing the algorithm in precisely the same way as before gave poor results. The reason is due to clustering of the images into very distinct groups. When the user is presented with two images, both very different from his target, his answer is virtually random. For instance, if he is searching for a forest, how does he choose between a calm sea and a brick wall? Yet for certain metrics in $\mathcal{D}$, one of these might be much farther from a potential target than the other and hence the feedback is misleading. Consequently, we introduced a third option for the user: Choose neither image. In this case, we simply modify the posterior distribution by assigning zero mass to the two displayed images and renormalizing. The mean search time is then reduced to approximately 20 queries (including the third option). In view of the exhaustion factor, the small size of the database and the fact that the mean search time for a totally random search is $\sum_{k=1}^{83} k\frac{2}{166} = 42$, this result remains poor.

Basically, there is an inefficient, transient search for the right "cluster" - the one containing the target - and an efficient, within-cluster search rather similar to that with polygons. One way to diminish the transient portion is to increase the number of images presented at each iteration. We displayed four images, allowing the user to select either no images or the ones closest to his target. This is equivalent to a superposition of binary comparisons and can be directly accommodated by the comparison search. The mean search time is then reduced to 11.

9 Conclusions

The experiments with real images are entirely preliminary. Perhaps a more "coarse-to-fine" search is necessary, where the database is organized hierarchically and the search proceeds in corresponding stages. It may be that two stages are enough, an initialization along the lines in [5] or with nearest-neighbors, and then entropy-based comparison search. Also, the uniform distribution over metrics is highly inefficient; as

with polygons, there is likely to be a substantial gain with dedicated distributions. Indeed, preliminary experiments in this direction, and with large image databases, are encouraging.

Our principal conclusion is that there is a sharp difference in the way performance depends on model complexity in simulations and in experiments with people. In the former case, when the answers are simulated according to the model, efficiency declines as complexity grows; the more randomness, the worse the performance. This is to be expected and is consistent with conventional patterns in inductive learning and nonparametric statistics, where complexity comes at a price. However, when human beings provide the answers, the efficiency grows with complexity; in particular, the most general model with dedicated distributions consistently performs better than simpler models. This seems to be due to "allowing more room" for variation and subjectivity in human decisions. Finally, it remains to be seen whether any of the models presented here is sufficiently general to accommodate very large databases.

Acknowledgements

Much of this work was carried out at the University of Massachusetts during an internship for Roland Moquet and Matthieu Tisserand supported by Ecole Polytechnique. We would like to thank Matthieu Tisserand for invaluable contributions and Chahab Nastar for introducing us to the subject and providing a state-of-the-art platform for testing our ideas, namely the Surfimage retrieval system in Project IMEDIA at INRIA.

Références

[1] L. Breiman, J. Friedman, R. Olshen and C. Stone, *Classification and Regression Trees*, Wadsworth, Belmont, CA., 1984.

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

[3] I. Cox, M. Miller, T. Minka and P. Yianilos, An optimized interaction strategy for Bayesian relevance feedback, In *Proc. Computer Vision and Pattern Recognition (CVPR '98)*, Santa Barbara, June 1998.

[4] T. Minka and R. Picard, Interactive learning using a society of models, *Pattern Recognition*, 30(4), 1997.

[5] C. Nastar, M. Mitschke and C. Meilhac, Efficient query refinement for image retrieval, In *Proc. Computer Vision and Pattern Recognition (CVPR '98)*, Santa Barbara, June 1998.

[6] M. Tisserand and R. Moquet, A flexible algorithm for image retrieval, Technical Report, Ecole Polytechnique, Palaiseau, France, June 1998.