Information retrieval in multimedia databases using relevance feedback algorithms.
Applying logistic regression to relevance feedback in image retrieval systems.

Autor:

Pedro Diego Zuccarello {pzuccar@dcom.upv.es}

Directores:

Teresa León {teresa.leon@uv.es} – Guillermo Ayala {guillermo.ayala@uv.es} (Depto.de Estadística e Investigación Opertiva – Universitat de Valencia)

Co-director en la ETSIT:

Alberto González Salvador {agonzal@dcom.upv.es}

Fecha de entrega:

7 de septiembre de 2007

Abstract — This master tesis deals with the problem of image retrieval from large image databases. A particularly interesting problem is the retrieval of all images which are similar to one in the user's mind, taking into account his/her feedback which is expressed as positive or negative preferences for the images that the system progressively shows during the search. Here, a novel algorithm is presented for the incorporation of user preferences in an image retrieval system based exclusively on the visual content of the image, which is stored as a vector of low-level features. The algorithm considers the probability of an image belonging to the set of those sought by the user, and models the logit of this probability as the output of a linear model whose inputs are the low level image features. The image database is ranked by the output of the model and shown to the user, who selects a few positive and negative samples, repeating the process in an iterative way until he/she is satisfied. The problem of the small sample size with respect to the number of features is solved by adjusting several partial linear models and combining their relevance probabilities by means of an ordered weighted averaged (OWA) operator. Experiments were made with 40 users and they exhibited good performance in finding a target image (4 iterations on average) in a database of about 4700 images.

Contents

Chapter 1

Introduction and related work

1.1 Introduction

The last few years have witnessed an increasing amount of pictorial information in different digital formats. Thus large image databases raise the need to retrieve relevant data efficiently. In this framework, content-based image retrieval systems (CBIR) are one of the most promising techniques for retrieving multimedia information [20], [15], [11]. CBIR systems are thought of as an improvement on traditional image retrieval systems based on textual information such as keywords. The new CBIR systems take advantage of valuable digital information held by the image itself. Visual features related to color, shape and texture are extracted in order to describe the image content [7]. The main drawback of textual image retrieval systems, that it, the annotator dependency, would be overcome in pure CBIR systems. Several papers have been published trying to integrate both approaches: textual and content-based image retrieval ([4] [26]).

Image features are a key aspect of any CBIR system. A general classification can be made: low level features (color, texture and shape) and high level features (usually obtained by combining low level features in a reasonably predefined model). High level features have a strong dependency on the application domain, therefore they are not usually suitable for general purpose systems. This is the reason why one of the most important and developed research activities in this field has been the extraction of good low level image descriptors. Obviously, there is an important gap between these features and human perception (a semantic gap). For this reason, different methods (mostly iterative procedures) have been proposed to deal with the semantic gap [18]. In most cases the idea underlying these methods is to integrate the information provided by the user into the decision process. This way, the user is in charge of guiding the search by indicating his/her preferences, desires and requirements to the system. The basic idea is rather simple: the system displays a set of images (resulting from a previous search); the user selects the images that are relevant (desired images) and rejects those which are not (images to avoid) according to his/her particular criterion; the system then learns from these training examples to achieve an improved performance in the next run. The process goes on iteratively until the user is satisfied.

The iterative algorithms which, in order to improve the set resulting from a query, require the user to enter his/her preferences in each iteration, are called relevance feedback algorithms [27].

These algorithms have been shown to provide a dramatic boost in retrieval system performance. Being part of this mainstream, this paper presents a new algorithm for relevance feedback in image databases based on logistic regression models.

A query can be seen as an expression of an information need to be satisfied. Any CBIR system aims at finding images relevant to a query and thus to the information need expressed by the query. The relationship between any image in the database and a particular query can be expressed by a relevance value. This relevance value relies on the user-perceived satisfaction of his/her information need. The relevance value can be interpreted as a mathematical probability (a relevance probability). In this paper a relevance probability $\pi(I)$ is a quantity which reflects the estimation of the relevance of the image I with respect to the user's information needs. Initially, every image in the database is equally likely, but as more information on the user's preferences becomes available, the probability measure concentrates on a subset of the database. The iterative relevance feedback scheme proposed in the present paper is based on logistic regression analysis for ranking a set of images in decreasing order of their evaluated relevance probabilities.

Logistic regression is based on the construction of a linear model whose inputs, in our case, will be the image characteristics extracted from the image I and whose output is a function of $\pi(I)$. The order of this model must be chosen, which is a key point. In logistic regression analysis, one of the key features to be established is the order of the model to be adjusted. The order of logistic regression model, the number of image characteristics, and the number of relevant (positive/negative) images the user is prompted to select, are strongly related. The order of the model must be in accordance with the reasonable amount of feedback images requested from the user. For example, it is not reasonable for the user to select 40 images in each iteration; a feedback of 5/10 images would be acceptable. This requirement leads us to group the image features into n smaller subsets, each consisting of semantically related characteristics. The outcome of this strategy is that n smaller regression models must be adjusted: each sub-model will produce a different relevance probability $\pi_k(I)$ $(k = 1 \dots n)$. We then face to the question of how to combine the $\pi_k(I)$ in order to rank the database according to the user's preferences. We tackled this problem by making use of the so-called OWA (*ordered weighted averaging*) operators which were introduced by Yager in 1988 [25] and provides a consistent and versatile way of aggregating multiple inputs into one single output.

Section 1.2 describes related work concerning relevance feedback procedures. In chapter 2 theoretical concepts such as image feature extraction, generalized linear models or ordered weighting averaging aggregation, needed to understand our new approach to relevance feedback for CBIR systems are explained. In chapter 3, all the previously explained concepts are put together into a new relevance feedback procedure. After that, in chapter 4 we present experimental results which evaluate the performance of our technique using real-world data. Finally, in chapter 5 we extract conclusions and point to further work.

1.2 Related work

Relevance feedback is a term used to describe the actions performed by a user to interactively improve the results of a query by reformulating it. An initial query formulated by a user may not fully capture his/her wishes. This is due to several reasons: the complexity of formulating the query, lack of familiarity with the data collection procedures, or inadequacy of the available features. Users then typically change the query manually and re-execute the search until they are satisfied. By using relevance feedback, the system learns a new query that better captures the user's need for information.

In recent years, several methods have been developed to guide the searching process in a retrieval system. All these techniques can be roughly classified into two different groups:

- Query point movement: the method of the query point movement approach is to construct a new query point that is supposed to be close to the relevant results and far from those which are non-relevant. The best-known approach for achieving query point movement is based on a formula initially developed by Rocchio in the context of textual information retrieval [17] .
- Reshaping distance functions: the objective of this approach is to modify the distance function in such a way that it can improve the query results according to the user's criterion.

A procedure belonging to the query point movement group was proposed by Ciocca and Schettini [3], who introduce a very simple algorithm for computing a new query point $\mathbf{Q}$ that can better represent the images of interest to the user. The procedure takes the set of relevant images the user has selected and computes a new point based on the standard deviation of the features used, computed separately one by one. Obviously, this ignores the dependency between image features, which is particularly important when they are values sampled from continuous functions.

Another implementation of point movement strategy consists of using the Bayesian method. Cox uses an adaptive Bayesian scheme which incorporates user preferences by means of a model of the user [5]. This model, together with the prior, gives rise inductively to a probability distribution on the event space. Experiments show that retrieval performance can be improved considerably by using such relevance feedback approaches. Relevance feedback has been also considered as a Bayesian classification problem by [8].

Yet another approach was taken by Rui et al., who propose an interactive retrieval approach which takes into account the user's high level query and perception subjectivity by dynamically updating certain weights [18]. Specifically, in this paper the images are represented by vectors of weights in the space of low level characteristics; these weights capture the importance of components within a vector as well as their importance across different vectors over the entire data set. The system then uses relevance feedback to update queries so as to place more weight on relevant elements and less on irrelevant ones.

The system *MindReader* proposed by Ishigawa uses a method that combines ideas from the query-point movement and axis re-weighting [12]. The goal of this method is, given n images selected by a user in the relevance feedback step, to compute the coefficients of a distance function (namely, the distance matrix M) at the same time as the best query point q that represents the n images selected by the user in the relevance feedback procedure. By solving a minimization problem on the parameter estimation process, the authors conclude that the best

M matrix (restricted to diagonal matrices only) is given by $m_{ij} \propto \frac{1}{\sigma_j^2}$, σ_j being the standard deviation of the jth vector component.

Unlike the present paper, our previous work concerning relevance feedback CBIR algorithms was focused on a Bayesian strategy [6]. We followed the idea of modeling user preferences as a probability distribution. The person manifests his/her preferences on the set resulting from a query. The chosen images are considered as a sample. The previous information is modeled as the prior distribution and the choices are incorporated into the posterior distribution. The main advantage of our approach was to specifically work with a prior distribution such that the posterior distribution belongs to the same conjugate family of distributions.

Chapter 2

Previous theoretical concepts

In this chapter, different elements used in the novel proposed algorithm will be presented: image descriptors, how the relevance probabilities are calculated and the aggregation operators used to combine the different relevance probabilities.

2.1 Visual features

This section deals with the low level features the system uses for predicting human judgment of image similarity. Most of the features used are very simple, since our main goal is not to test the features as such, but only to use them as a tool to evaluate the new relevance feedback procedure. The relevance feedback methodology we have developed can be applied without changes to any image indexing and retrieval methods, even certainly a different set of features will give different results. With respect to the grouping of the features in smaller subsets, it is a resource to be able to apply logistic regression with a very small sample size. The results could change, too, by using a different grouping scheme.

Our system can currently work with different features that are obtained by preprocessing each image in the database. Amongst these characteristics we can mention those which represent chromatic information and those related to textures present in the image:

Color representation: the current version of the system incorporates as chromatic information of the image:

- A histogram of the HS (Hue, Saturation) values of the image pixels: these values are obtained after conversion to HSV color space and quantization into $(H \times S) = (10 \times 3) = 30$ color bins. This 2D histogram is flattened by rows (H-component) giving a vector of 30 features.

Texture representation: the system currently works with information about textures in the image. This information is embodied as:

- The granulometric cumulative distribution function. A granulometry is defined from the morphological opening of the texture using a convex and compact subset containing the origin as structuring element [21]. In our case we have used a horizontal and a vertical segment as the structuring elements.

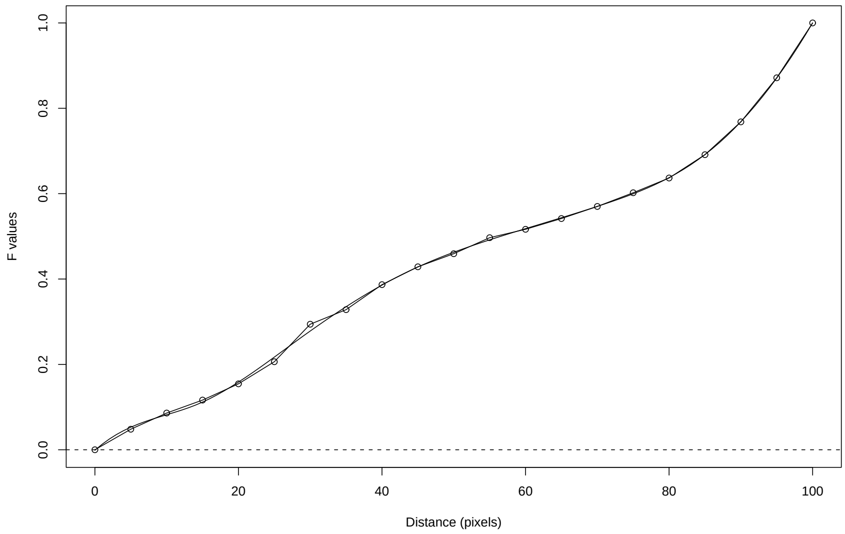


Figure 2.1: Granulometry distribution function F together with its approximation with a ten coefficient B-spline base

In this paper we have approximated the cumulative granulometric function sampled from 0 to 100 pixels at intervals of 5 pixels by using a spline basis with 10 basis functions. Fourier basis is more appropriate for periodic functions, which is not the case here. Wavelets, on the other hand, are especially appropriate when the function presents intervals of high variation or even discontinuities together with relatively smooth areas. Since our functions are continuous, smooth and increasing, a spline basis can give a highly accurate representation; moreover, as splines are functions of local support, their coefficients are related with particular parts of the function support, which is appropriate in cases like this in which the value of the granulometry has a direct relationship to the size of the structuring element, which is in turn related to the size of the particles in the texture of the image. Figure 2.1 shows a typical granulometry distribution function F sampled from 0 to 100 pixels at intervals of 5 pixels (with circles) together with its approximation with a ten coefficient B-spline basis f_b (smooth line). As can be appreciated visually, the fit is very good; numerically, the error can be calculated as

$$Error = \frac{1}{N}\sqrt{\sum_{i=1}^{N} \mid F(x_i) - F_b(x_i) \mid^2}.$$

N being the number of points and F_b the B-spline approximation of F; error is $2.6 \cdot 10^{-5}$ for this case; as a comparison, the mean value of F is 0.438.

2.2 Generalized linear models and CBIR

Logistic regression models are a particular case of generalized linear models (GLMs) [1]. GLMs extend ordinary regression models to encompass non-normal response distributions and modeling functions of the mean. All GLMs have three different components: the random component, the systematic component, and the link function.

2.2.1 The random component

The random component identifies a response variable Y and assumes a probability distribution for it. For a sample of size t, denote the observations on the response variable Y by $(Y_1, \ldots, Y_t)$. The GLMs used in this work treat $Y_1, \ldots, Y_t$ as independent. The random component of a GLM consists of identifying the response variable Y and selecting a probability distribution for $(Y_1, \ldots, Y_t)$. For example, in many cases, the potential outcomes for each observation Y_j are binary values. In these cases a binomial (or, for $t = 1$, Bernoulli) distribution could be assumed. In some other applications, each response observation is a nonnegative count, such as number of accidents in a certain roadway. In these cases, it is very common to use a Poisson distribution for the random component. If each observation is continuous, such as the pressure of a certain fluid, a normal random component may be assumed.

2.2.2 Systematic component

Let us denote the expected value of Y by $\mu = E(Y)$. In a GLM, the value of μ varies according to the values of certain variables related to the particular problem being analyzed and modeled. For example, if Y is the weight of a person, the value of μ can be related to the height of the person, or to his/her food ingesta (amount of fat, vegetables, sugar, etc., that the person eats). These variables are called explanatory variables, and are determined by the systematic component of the GLM.

Then, the mathematical expression of the model is:

$$g(\mu) = \alpha + \beta_1 x_1 + \ldots + \beta_p x_p, \tag{2.1}$$

where the function $g(.)$ is the so called link function and is usually determined according to the selected probability distribution for Y (more details will be given in section 2.2.3), $(\alpha, \beta_1, \ldots, \beta_p)$ are linear coefficients whose value is to be determined (more details on the computation of these coefficients are given in section 2.2.4), and $(x_1, \ldots, x_p)$ are the explanatory variables determined by the systematic component of the model. The right hand side of equation 2.1 is called the linear predictor.

2.2.3 The link function

The link function is the link between the random and systematic components. We already introduced this function as $g(\mu)$ in equation 2.1.

The simplest possible link function has the form $g(\mu) = \mu$. This models the mean directly and is called the identity link:

$$\mu = \alpha + \beta_1 x_1 + \ldots + \beta_p x_p. \tag{2.2}$$

This is the form of ordinary regression models for continuous responses.

Other links permit the mean to be nonlinearly related to the predictors, for instance $g(\mu) = log(\mu)$:

$$log(\mu) = \alpha + \beta_1 x_1 + \ldots + \beta_p x_p. \tag{2.3}$$

In this case, the GLM is usually called loglinear model. Loglinear models are appropriate when μ cannot be negative, such as with count data.

The link function $g(\mu) = log(\mu/[1-\mu])$ models the log of an odds. This is called the logit link. It is appropriate when μ is between 0 and 1, such as a probability. These type of GLMs are called logistic regression models, or logit models:

$$logit(\mu) = log(\mu/[1-\mu]) = \alpha + \beta_1 x_1 + \ldots + \beta_p x_p. \tag{2.4}$$

Each potencial probability distribution for the random component has one special function of the mean that is called its natural parameter. For the normal distribution, it is the mean itself (equation 2.2). For the Poisson distribution, the natural parameter is the log of the mean (equation 2.3). For the binomial distribution, the natural parameter is the logit of the success probability (equation 2.4). The link function that uses the natural parameter as $g(\mu)$ in the GLM is called the canonical link. Though other links are possible, in practice the canonical links are most common.

2.2.4 Application of logit GLM to CBIR

At each iteration, a sample of n images from the whole image database is evaluated by the user selecting two sets of images: the examples or positive images and the counter-examples or negative images. Let us consider the (random) variable Y giving the user evaluation where $Y = 1$ means that the image is positively evaluated and $Y = 0$ means a negative evaluation.

Each image in the database has been previously described by using low level features in such a way that the j-th image has the k-dimensional feature vector x_j associated. Our data will consist of (x_j, y_j), with $j = 1, \ldots, n$, x_j is the feature vector and y_j the user evaluation (1= positive and 0= negative). The image feature vector x is known for any image and we intend to predict the associated value of Y. The natural framework for this problem is the logit model. Therefore, in this work, a logistic regression model will be used, where $P(Y = 1 \mid x)$ i.e. the probability that $Y = 1$ (the user evaluates the image positively) given the feature vector x, is related with the systematic part of the model (a linear combination of the feature vector) by means of the logit function (equation 2.4). Most statistical software has the facility to fit GLMs. Logistic regression is the most important model for categorical response data. We can rewrite equation 2.4 as:

$$logit[\pi(x)] = \alpha + \beta_1 x_1 + \ldots + \beta_p x_p, \tag{2.5}$$

where $\pi(x) = P(Y = 1 \mid x)$ is the expected value of the binary response value Y conditioned to the values $x_1, \ldots, x_p$ of the explanatory variables.

The model can also be stated directly specifying $\pi(x)$ as

$$\pi(x) = \frac{\exp(\alpha + \beta_1 x_1 + \ldots + \beta_p x_p)}{1 + \exp(\alpha + \beta_1 x_1 + \ldots + \beta_p x_p)}. \tag{2.6}$$

The parameter β_i refers to the effect of x_i on the log odds that $Y = 1$, controlling the other x_j. The model parameters are obtained by maximizing the likelihood function given by

$$l(\beta) = \prod^{n} \pi(x_i)^{y_i} [1 - \pi(x_i)]^{1-y_i}. \tag{2.7}$$

The maximum likelihood estimators (MLE) of the parameter vector $\beta = (\alpha, \beta_1, \ldots, \beta_p)$ are calculated by using an iterative method. A detailed explanation can be found in [2] (pp. 192). We have used the public domain statistical program R [16]. In particular, the function *glm* from the package *MASS* [23]. An additional advantage of using linear models is that, if each feature has a known sematic meaning, the value of the parameter associated to it has to do with its importance for the viewer, and the width of its confidence interval might be consider to be related with the certainty with which the viewer uses that feature. Nevertheless, as we explain in the next paragraph, in our case the sematic meaning is not clearly associated to a particular parameter, but to a group of them (color parameters, texture parameters, etc.).

In the first steps of the procedure, we have a major difficulty when having to adjust a global regression model in which we take the whole set of variables into account, because the number of images (the number of positive plus negative images chosen by the user) is typically smaller than the number of characteristics. In this case, the regression model adjusted has as many parameters as the number of datum and many relevant variables could be not considered. On the other hand it is not realistic to ask the user to make a great number of positive and negative selections from the very beginning; therefore we think that the difficulty cannot be avoided in this way. In order to solve this problem, our proposal is to adjust different smaller regression models: each model considers only a subset of variables consisting of semantically related characteristics of the image. Consequently, each sub-model will associate a different relevance probability to a given image x, and we face the question of how to combine them in order to rank the database according to the user's preferences. We can see this question as an information fusion problem.

2.3 Ordered weighting averaging

Let us denote as $\pi_1(x), \pi_2(x), \ldots, \pi_n(x)$ the different relevance probabilities associated with a given image x. Each one of them has been obtained separately by using different regression models and we need to associate a final probability $\pi(x)$ by aggregating the information provided by each $\pi_j(x)$, $(j = 1 \ldots n)$. Mathematical aggregation operators transform a finite number of inputs into a single output and play an important role in image retrieval. In [22]the authors compare the effect of 67 operators applied to the problem of computing the overall image similarity, given a collection of individual feature similarities. Their results show how important for retrieval performance the choice of the aggregation operator is. A comprehensive overview, as well as the classification, of mathematical aggregation operators can be found in [19]. We

Table 2.1: Illustrating examples of OWA aggregation values.

W	$f(a_1,\ldots,a_n)$
$(1,0,\ldots,0)$	$\max_i a_i$
$(0,0,\ldots,1)$	$\min_i a_i$
$(\frac{1}{n},\frac{1}{n},\ldots,\frac{1}{n})$	$\frac{1}{n}\sum_{j=i}^{n} a_i.$

have not used any of the 67 operators reviewed; instead, we decided to use the so-called ordered weighted averaged (OWA) operators to aggregate our relevance probabilities. These operators were introduced in [25]. Since then they have been successfully applied in different areas: decision making, expert systems, neural networks, fuzzy systems and control, etc. An OWA operator of dimension n is a mapping $f : \mathbb{R}^n \to \mathbb{R}$ with an associated weighting vector $W = (w_1,\ldots,w_n)$ such that $\sum_{j=1}^{n} w_j = 1$ and where $f(a_1,\ldots,a_n) = \sum_{j=1}^{n} w_j b_j$ where b_j is the j-th largest element of the collection of aggregated objects $a_1,\ldots,a_n$. The particular cases shown in table 2.1 can better illustrate the idea underlying OWA operators.

Notice that no weight is associated with any particular input; instead, the relative magnitude of the input decides which weight corresponds to each input. In our application, the inputs are relevance probabilities and this property is very interesting because we do not know, a priori, which set of visual descriptors will provide us with the *best* information.

As OWA operators are bounded by the max and min operators, Yager introduced a measure called *orness* to characterize the degree to which the aggregation is like an *or* (max) operation:

$$\text{orness}(W) = \frac{1}{n-1}\sum_{i=1}^{n}(n-i)w_i. \tag{2.8}$$

Note that for $W = (1,0,\ldots,0), \text{orness}(W) = 1$, for $W = (0,0,\ldots,1), \text{orness}(W) = 0$ and for $W = (\frac{1}{n},\frac{1}{n},\ldots,\frac{1}{n}), \text{orness}(W) = 0.5$ This author also introduced the concept of *dispersion* or *entropy* associated with a weighting vector:

$$\mathit{Disp}(W) = \sum_{i=1}^{n} w_i \ln w_i. \tag{2.9}$$

$\mathit{Disp}(W)$ tries to reflect how much of the information in the arguments is used during an aggregation based on W.

Clearly, the vector of weights W can be pre-fixed, but a number of approaches have also been suggested for determining it according to different criteria. For instance, in [9] an algorithm which can be used to learn the weights from an observation of performance by others. One of the first methods developed was proposed by O'Hagan [14]. It provides us with the vector of weights for a given level of orness (optimism) which maximizes their entropy:

$$W = \mathit{argmax}\sum_{i=1}^{n} w_i \ln w_i, \quad \text{subject to} \begin{cases} \alpha = \frac{1}{n-1}\sum_{i=1}^{n}(n-i)w_i, \\ \sum_{i=1}^{n} w_i = 1, w_i \in [0,1]. \end{cases} \tag{2.10}$$

Roughly speaking, when the maximum entropy vector of weights for a prefixed orness is calculated, the set of weights most similar to a discrete uniform distribution is obtained, in the sense that each weight tends to achieve its maximum value while the sequence of weights increases or decreases (depending on the orness value). By this we mean that if $W = (w_1, \dots, w_n)$, then $w_1 \geq w_2 \geq \dots \geq w_n$, or $w_1 \leq w_2 \leq \dots \leq w_n$. This problem is not computationally easy to solve. Fuller and Majlender [10] have obtained the analytical expression of the maximum entropy weights. Their practical procedure involves the solution of a polynomial equation; however, the unique root of a rather flat polynomial is numerically difficult to obtain.

We have used a simpler procedure to generate the set of weights $W = (w_1, \dots, w_n)$. They are obtained as a mixture of the binomial $Bi(n-1, p)$ and the discrete uniform probability distributions. A principal advantage of this choice is its flexibility and simplicity: the weights are easily obtained and are also easy to interpret; however, the main reason for choosing this procedure is that our practical experiments have shown that it works well for our case.

Notice that OWA operators with many of the weights close to their highest values will be *or-like* operators (orness(W) ≥ 0.5), while those operators with most of the weights close to their lowest values will be *and-like* operators (orness(W) ≤ 0.5). The sequence of weights that we can obtain as a mixture of a binomial and a discrete uniform distribution is not necessarily increasing or decreasing and the combination of the values for the parameter p of the binomial probability distribution and the mixture parameter λ allows us to obtain a parametric family of weights easily.

Some properties

Proposition 1. Let $W = (w_1, \dots, w_n)$ and $W' = (w'_1, \dots, w'_n)$ be two vectors of weights such that orness $(W) = \alpha$ and orness $(W') = \beta$ with $\alpha, \beta \in [0, 1]$, then $\forall \lambda \in [0, 1]$ we have the orness of $(\lambda W + (1-\lambda)W') = \lambda\alpha + (1-\lambda)\beta$, where $\lambda W + (1-\lambda)W' = (\lambda w_1 + (1-\lambda)w'_1, \dots, \lambda w_n + (1-\lambda)w'_n)$

The proof follows from the definition of orness.

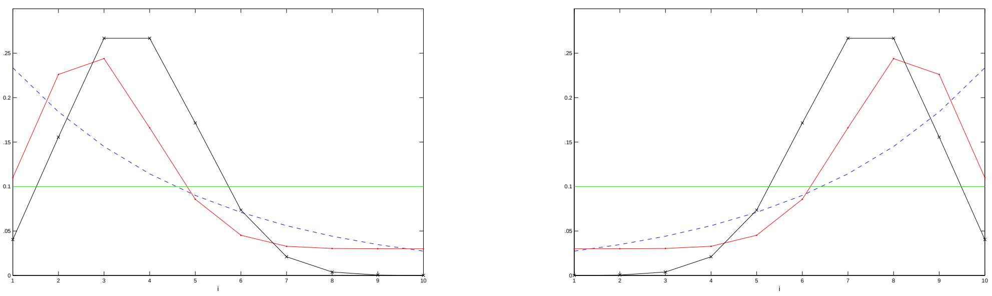
Proposition 2. Let $W = (w_1, \dots, w_n)$ be a vector of weights such that $w_i = \pi_{i-1} = \binom{n-1}{i-1}(1-\alpha)^{i-1}\alpha^{n-i}$, then orness$(W) = \alpha$.

Proof. Notice that the components of the vector of weights are the n probabilities of a binomial probability distribution with parameters $n-1$ and $p = 1-\alpha$, therefore

$$\text{orness}(W) = \frac{1}{n-1}\sum_{i=1}^{n}(n-i)\pi_{i-1} = \frac{1}{n-1}(n - \sum_{j=0}^{n-1}(j+1)\pi_j) = \frac{1}{n-1}(n-(n-1)(1-\alpha)-1) = \alpha. \quad (2.11)$$

We can use these results to obtain different sets of weights for a given orness.

Corollary 1. Let $W = (w_1, \dots, w_n) = (\lambda\pi_0 + (1-\lambda)\frac{1}{n}, \dots, \lambda\pi_{n-1} + (1-\lambda)\frac{1}{n})$, where $\pi_j = P(X = j)$ with $X \sim B(n-1, p)$, then orness$(W) = \lambda(1-p) + (1-\lambda)0.5$



(a) (b)

Figure 2.2: Aggregation weights with n=10 obtained using the Fuller and Majlender method (blue dashed line) with orness α, probabilities of a $Bi(n-1, 1-\alpha)$ distribution (x-marked line), mixture with parameter λ of a $Bi(n-1, p)$ distribution and a discrete uniform (dotted line), and discrete uniform (continuous green line). (a) $\alpha = 0.7$, $\lambda = 0.7$ and $p = 0.2143$, (b) $\alpha = 0.3$, $\lambda = 0.7$ and $p = 0.7857$.

A direct consequence of this corollary is that, for a given orness α, we can construct a vector of weights as a mixture of binomial and discrete uniform probabilities. The relationship between α, λ, and p can be expressed as:

$$2\alpha - 1 = \lambda(1 - 2p), \tag{2.12}$$

i.e. the set of pairs $\{(p, \lambda) : 2\alpha - 1 = \lambda(1 - 2p),\ \lambda \in (0, 1]\}$ characterizes the different mixtures which give rise to the different vectors of weights for a given orness α.

We can distinguish three cases.

1. If $\alpha < 0.5 \to 2\alpha - 1 < 0$ (and $\lambda > 0$) $\to 1 - 2p < 0 \to p > 0.5$. Moreover, as $\lambda \leq 1$, we have that $1 - 2\alpha \leq 2p - 1$ and then $p \geq 1 - \alpha$ with $\lambda \in (0, 1]\}$

2. Analogously, if $\alpha > 0.5$ we obtain $p \leq 1 - \alpha$ with $\lambda \in (0, 1]$.

3. Finally, if $\alpha = 0.5$, then $\lambda = 0$ and $p \in (0, 1)$ or $p = 0.5$ and $\lambda \in (0, 1]$.

Figure 2.2 illustrates a comparison of the aggregation of weights for $n = 10$ obtained with the above-mentioned methods: Fuller and Majlender, a binomial probability density function, and a mixture of a binomial probability density function and a discrete uniform. In Figure 2.2(a) we can see that the maximum entropy vector of weights might attach too great an importance to the biggest input, and also that the binomial probability distribution is "too extreme", although, the addition of the uniform part of the mixture "smoothes" its shape.

Chapter 3

Novel relevance-feedback procedure

So far we have made a detailed description of the different theoretical components of our novel search strategy. It is now time to explain how we combine them into an efficient relevance feedback algorithm.

Let us assume a collection of images (the database) where a set of image features has been computed off-line for each image in the collection. Our particular choice of features has been: first, the 30 values of the HS-histogram bins; then, the granulometric cumulative distribution function is calculated for two different structuring elements: a vertical and a horizontal line, and sampled at intervals of 5 pixels between 0 and 100. This gives 20 values for each structuring element. Instead of using these values as raw data we approximate the function by expressing it in a B-spline basis of 10 basis functions and taking as features the coefficients. This add up to 10 coefficients per structuring element, so we have a total of 20, which added to the 30 values of the histogram results in a final vector of 50 features per image. See detailed explanation in section 2.1. With respect to the grouping of the characteristics which are semantically related to apply the approach based on several sub-models, we have considered 10 groups, each one made by 5 consecutive characteristics. Notice that 6 of these groups are related to colour values and 4 to texture values. No group contains both types of characteristics. In section 4 we discuss through an experiment the influence of having more or less groups, with the condition that each group contains only one type of feature, either colour or texture. The choice of these particular groups has been motivated by common sense since it seems natural to group together color features, and in a different group from texture ones. Nevertheless, the choice of appropriate groups is an important and interesting topic because of two reasons: first, it may improve the performance of the retrieval algorithms and second, it may make apparent hidden groups that are into the viewer's mind (therefore, properly semantic) which he/she is unable to express in words or even realize of its existence. This is a possible line of research for future work or to be considered by people involved in artificial intelligence, concretely knowledge elicitation.

Let us also assume that the images are initially randomly ranked. Each iteration of the relevance feedback algorithm changes the ranking of the images according to a given set of data. By data we mean a user selection of positive and negative relevant images, and a set of aggregation weights.

An schematic description of the procedure is as follows:

Initialization: Images are randomly ranked.

Input parameters: Positive and negative relevant images are selected from amongst the whole collection. Let $\mathbf{I_P}$ be the set of positive samples, and $\mathbf{I_N}$ the set of negative samples. Let $W = (w_1, \ldots, w_n)$ also be the set of aggregation weights, where n is the number of relevance probabilities (outputs of the different logistic regression models) to be combined.

Logistic regression model: Using inputs selected in the previous step, several logistic regression models are fitted. Such models are applied to each image I_j in the database, obtaining their respective relevance probabilities, $(\pi_1(I_j), \ldots, \pi_n(I_j))$.

Aggregation and ranking: In order to obtain a unique relevance value, the relevance probabilities $\pi_1(I_j) \ldots \pi_n(I_j)$ should now be aggregated using the previously selected weights W (see section 2.3 for a detailed description of OWA aggregation operator). Images are ranked according to the computed relevance values.

The numbers of positive and negative relevant images are not required to be equal. In the first iterations it is the usual case not to find many positive relevant images, therefore the number of images in set $\mathbf{I_P}$ is not very large. This is not the case for negative relevant selection, therefore the set $\mathbf{I_N}$ usually has many more images than $\mathbf{I_P}$.

When a user rejects an image by selecting it as negative, we assume that the user's wish for that particular image will not change at any point in the searching process. Therefore we have implemented a memory algorithm for the selection of negative relevant images. Negative selections are remembered through all iterations. In iteration r, the set of negative relevant images used as input for the logistic regression model, $\mathbf{I_N}$, is obtained as:

$$\mathbf{I_N} = \mathbf{I_N^r} \cup \mathbf{I_N^{prev}}, \tag{3.1}$$

where $\mathbf{I_N^r}$ is the set of negative images selected by the user in iteration r, and $\mathbf{I_N^{prev}}$ is a subset of randomly selected images from user negative selections made in iterations 1 to $r-1$. The probability of a certain image, I_i, belonging to $\mathbf{I_N^{prev}}$ is:

$$P(I_i) = \frac{i}{\sum_{q=1}^{r-1} qN_q}, \tag{3.2}$$

where I_i is each image selected as negative in iteration i, N_q is the number of negative selected images in iteration q, and r is the present iteration.

No memory was implemented for positive relevant selection; in each iteration the logistic regression models are fitted with just the positive choice of the current iteration. The rationale behind this is to allow the user to focus his/her search on progressively narrower sets and allow him/her to choose as negative samples images that were formerly positive, possibly because none of the images shown in former iterations were sufficiently similar to the target.

With respect to the choice of the orness value for the OWA aggregation operator, the user can choose it freely. He/she receives a previous explanation about its meaning: the idea is to start with a high orness value (allowing recalls of images similar only by one of the groups, which from his/her point of view means loosely similar) and decreasing later to a more restrictive choice (similarity by all of the groups) as the search proceeds and is more centered on really similar images. In most of the cases it was sufficient to make one change of the orness value during the search.

Chapter 4

Experimental results

4.1 Experimental setup

The main objective of our algorithm is to find an image which is similar to what the user may have in mind. Therefore, the first step in the design of the experiments would be to define what is understood by "similar". Unfortunately, this is not easy since it depends on the user, and the goal of the algorithm is precisely to capture that notion of similarity that each user has, which can also change between different queries. Consequently, the valid criterion of similarity appears to be the user's opinion. This would have introduced an external variable into the experiment that would have masked the main goal: an objective evaluation of the system as such. That is why we have chosen to use an approach in which a given image has to be found. The search is considered successful if the image is ranked within the first 16. This number is arbitrary but we have checked that 16 images shown side by side is a reasonable number to localize a particular one at a first sight.

Once the criterion for termination has been adopted, the experiment will be designed by showing several images to the user; a choice of 6 images (the same for all users) was selected from a database of about 4700. The pictorial database was assembled using some images obtained from the web and others chosen by the authors. These images are classified as belonging to different themes such as flowers, horses, paintings, skies, textures, ceramic tiles, buildings, clouds, trees, etc. even though the category is not used at all during the search. The 6 target images are in our experience, representative of different types and levels of difficulty. They are displayed in figure 4.1.

For each target image the search proceeds iteratively. In each iteration the user has to select some relevant images (similar to the target according to his/her judgment) and others significantly different from the target. The number of images of each type is left to the user, although two conditions must be fulfilled: at least one relevant and one irrelevant images must be selected and the total number of selections has to be greater than the maximum number of features jointly considered. For instance, if we group the features in groups of five then the minimum number of images evaluated has to be five per iteration.

The algorithm proceeds as explained in previous sections and shows the images in the database ordered according to their similarity to the target, in groups (screens) of 32 (the 16 more similar and the 16 more dissimilar). If the target appears in the first 16, it is considered to have

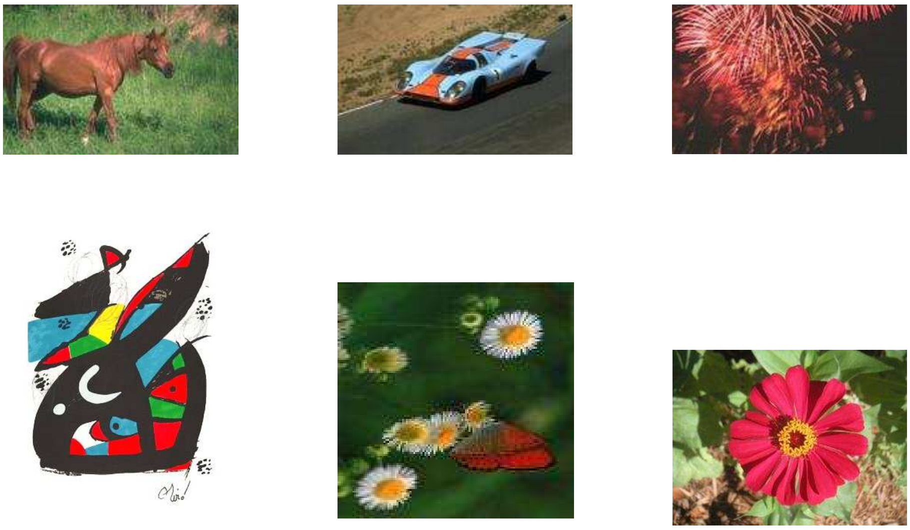


Figure 4.1: Target images used in experiments.

been found; otherwise the user can move backwards or forwards to see more images in rank order and a new iteration of choosing/search/showing begins.

All the experiments have been performed using a graphical user interface GVI(see figure 4.2), a system previously programmed by our research group that allows the easy introduction of new images, new features associated to each image and new similarity measures, being a powerful and flexible tool. GVIis oriented to an expert user with some basic knowledge of image processing and content-based retrieval. The creation of retrieval systems for multimedia data is a non-trivial problem, which could be facilitated with a system like this.

The variables we intend to analyze in our experiment are:

- The number of iterations used by each person to find the target. This is the most important variable for a system of this type.
- The number of iterations used to find an image, independently of the user. This is intended to evaluate if the nature of the image has an important influence on the result.
- The number of relevant and non relevant images chosen as a function of the image.
- The same as a function of the iteration. This should be related to the progressive convergence towards the target.
- The position of the target in the rank for each iteration.

All these variables were evaluated for 40 users who did the test for the 6 chosen images; amongst them there were computer programmers, mathematicians, children and one graphical designer, all of different ages and levels of exposure to computers.

A general graphical and numerical description is provided in section 4.2. Then, a non-supervised classification of the user behavior will be given in section 4.3.

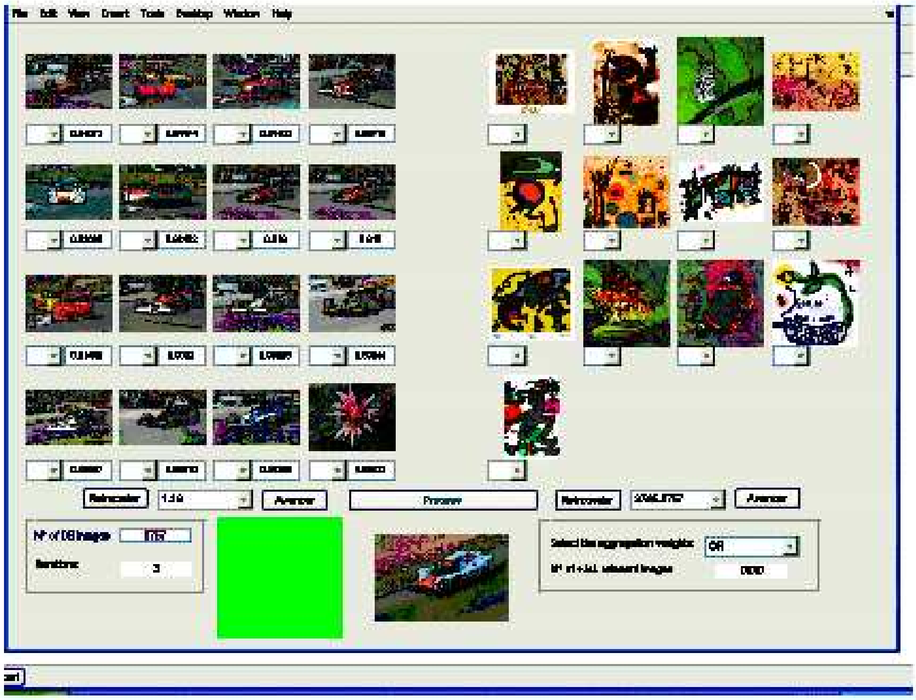


Figure 4.2: Graphical user interface used in the experiments.

4.2 Descriptives

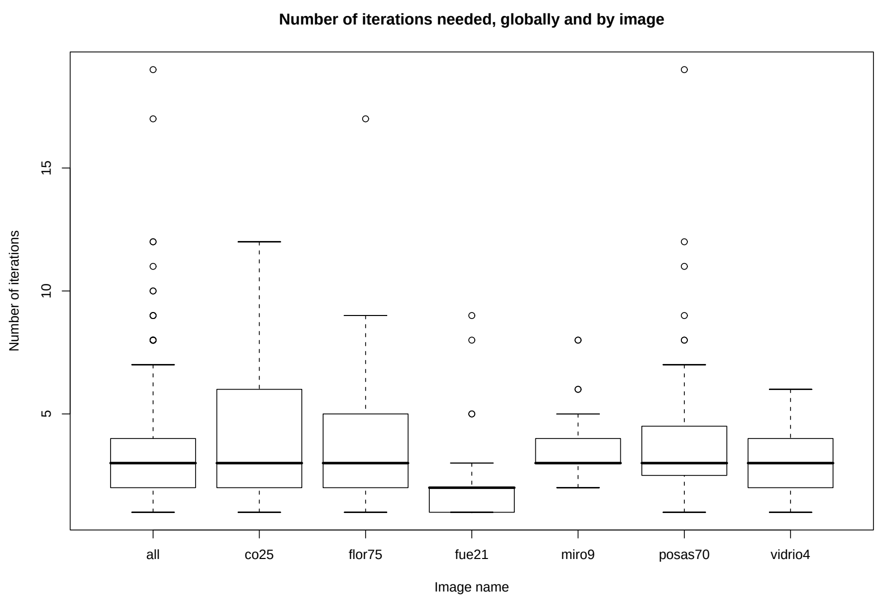
Firstly, a descriptive study of the previously indicated variables will be given. For each image and trial we consider the number of iterations, the mean rank of the target image and the mean number of positively and negatively chosen images. These means are calculated throughout the iterations of each trial. The description will be given in table 4.2 in terms of the first, second and third quartiles in the column under the heading “All”. It is noticeable that the number of iterations goes from 3.12 to 4.43, which can be considered to be quite small. We have not taken into account the time for performing the experiment because it depends on the degree of knowledge of the application that the user has. However, a typical trial is usually done by looking at three to four displays with a duration of 10 to 20 seconds per display, which adds up to a total time of around one minute per trial.

Concerning the mean rank, it should be noticed that in our experimental setup, the target image is randomly located in the last third of the database, i.e. its initial rank is greater than 3000. During the iterations the rank decreases until finishing before 16^{th}; in this process the quartiles of the rank go from 195 to 343, with a median of 274.

Regarding the number of evaluated images, as usual in these kind of procedures, the user does not select a great number, but consistently selects a greater number of negative samples (median of 8.22) than positive ones (median of 3.89).

Clearly, all these variables depend heavily on the behavior of each user. This will be carefully analyzed in the next subsection.

The dependence on the image will be evaluated by means of a multiple box plot displayed in figure 4.2. Two apparent facts have to be pointed out. The medians are very similar, except for image fue21 (a photograph of fireworks) which is the easiest one. However, the variability is clearly different between images. The largest box corresponds to image co25 (a car), but there are outliers in most of the data groups, suggesting that certain images are particularly difficult for certain users. Again, no common behavior appears to arise amongst all the users, which leads us to try a non-supervised classification of users.



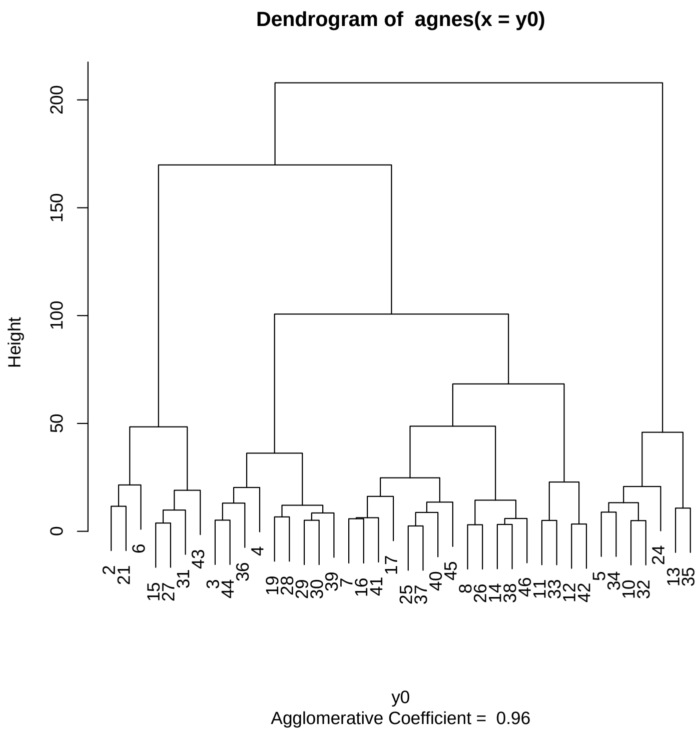


Figure 4.3: Dendrogram

4.3 Non-supervised classification

We will describe the trial corresponding to a given user by means of four variables: the mean number of iterations, the mean ranking of the image that we are looking for throughout the iterations, the mean number of images positively evaluated and the mean number of images negatively evaluated i.e. the experiment for a user is summarized as a four feature vector. The results observed in a trial are the combined consequence of the software tool to be evaluated and the characteristics of the particular user. It seems to us very interesting to find groups of similar users by taking into account the feature vector which has just been mentioned. Therefore, we have computed an agglomerative hierarchical clustering of the dataset to explore its structure. Figure 4.3 shows the corresponding dendrogram. At first sight, four different clusters are observed. In order to confirm this, we have also used a non hierarchical partitioning method. Partitioning methods require that the number of clusters, k, be given by the user; the clustering method which seems natural in this context is the partitioning around medoids. The first step is to find k representative objects or medoids from among the observations in the dataset. These observations should represent the structure of the data. The k clusters are constructed by assigning each observation to the nearest medoid. We have used the silhouette widths for assessing the best number of clusters.

Let us briefly recall the definitions of silhouette and silhouette plot in a graphical display, in which for each observation i, a bar is drawn, representing its silhouette width.

For each observation, i, the silhouette width, s(i), is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{4.1}$$

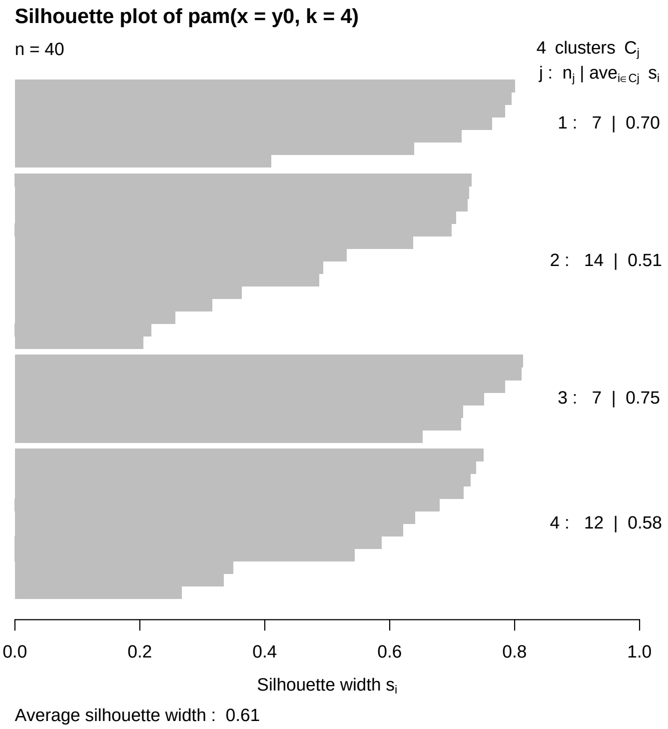


Figure 4.4: Silhouette observed using four groups

where a(i) is the average dissimilarity between i and all other points of the cluster to which i belongs, and b(i) is the dissimilarity between i and its neighboring cluster i.e. the nearest one to which i does not belong. Clearly, observations with a large s(i) are very well clustered, s(i) around 0 means that the observation is between two clusters, and observations with a negative s(i) are probably placed in the wrong cluster.

Moreover, the average silhouette allows us to select the best number of clusters: a clustering can be performed for several values of k, the number of clusters and then we choose the value of k with the largest overall average silhouette width. This value is equal to four in our case. The silhouette widths, together with the average silhouette width for each group and the (global) average silhouette width, can be found in Figure 4.4. Notice that the average silhouette width is equal to 0.61, suggesting that the structure of the dataset is quite well defined.

We have also performed a k-means analysis with $k = 4$ and found exactly the same clustering, this fact confirming the robustness of our results.

We would like to mention that these four clusters are determined both for the personality of the user and also the kind of experience that we have carried out. We are aware that our experiment is not the ideal one for testing our application but, in our opinion, it is not easy to improve on. The aim of our procedure is not to help the user retrieve a particular image in the dataset, as in the experiment. Instead, its aim is to help users find those images with the features that they have in mind.

Table 4.1 displays the medoids or representative individuals of the different clusters and some descriptive statistics for the four clusters. This table allows us to interpret the particular characteristics of the individuals in each cluster.

Table 4.1: Medoids

Medoid	Iterations	Rank	Positive	Negative
1	3.86	459.56	3.48	8.22
2	4.70	314.08	3.13	12.17
3	3.00	103.24	7.45	10.44
4	5.57	224.41	5.67	9.30

Table 4.2: Basic descriptives of the variables mean number of iterations, mean rank, mean of positively and negatively evaluated images.

		Group 1	Group 2	Group 3	Group 4	All
Iterations	1st Qu.	4.14	3.33	2.14	3.40	3.12
	Median	4.43	3.79	2.29	4.17	3.79
	3rd Qu.	4.86	3.74	3.40	4.43	4.43
Ranks	1st Qu.	419.2	288.6	85.6	195.3	195.3
	Median	448.9	318.0	103.2	222.0	274.0
	3rd Qu.	460.9	337.3	114.6	238.8	343.9
Positive	1st Qu.	3.30	3.18	3.29	3.65	3.29
	Median	3.48	3.77	4.79	5.12	3.89
	3rd Qu.	4.62	4.14	7.13	6.57	3.90
Negative	1st Qu.	5.27	5.80	7.20	6.65	5.74
	Median	5.70	8.46	8.97	8.09	8.22
	3rd Qu.	8.35	10.99	10.45	9.39	10.06

The users belonging to cluster 1 are the impatient ones: they choose just a few positive and a few negative images and, although they find their objective, they do it in the end or with ups and downs (large average position). On the other hand, the individuals in cluster 3 are the most patient: they find the target with the minimum number of iterations and the best average position; these users make a lot of positive selections and also many negative selections. The individuals clustered in 2 mark many images negatively but they only select a few positive images, their average position through the search is smaller than the average position for cluster 1. The users classified as cluster number 4 need more iterations to achieve their objective, and although they make many positive selections they are not very lucky in their choices.

In any case, we think that we have obtained quite good results considering that the average number of iterations in every cluster is very reasonable .

4.4 Influence of the grouping of characteristics

A potentially important point is the influence on the results of the way in which the components of the feature vector are grouped in sub-models to be fitted by the regression algorithm and it is reasonable to pose the question of what is better: few sub-models with a relatively large number of characteristics or more sub-models with a few components each. Apart from avoiding the mixing of feature of different nature (colour with texture, in our case) there is no a clear answer. An experiment has been done to enlight this issue by using groups of 3, 5 and 7 features to recall seven different images, being the other parameters as in former experiment, and equal for the three trials. Results in terms of number of trials to find the target can be seen in table 4.3.

Table 4.3: Number of iterations to recall an image depending on the size of the chosen sub-models.

Sub model size	Image number							Mean	St. dev.
	1	2	3	4	5	6	7		
3	4	10	1	1	6	2	6	4.28	3.30
5	4	2	1	7	4	1	5	3.43	3.30
7	1	3	2	5	6	4	8	4.14	2.41

The mean and standard deviation of the number of iterations appears to be slightly smaller for models of five characteristics than for the others, although given the sample size (7 images) this cannot be considered significant. This experiment has suggested us the convenience of choosing the five characteristics model, taking into account essentially the balance between the robustness of the model (in principle, bigger for larger groups) and the minimum number of images a user is asked to select.

Chapter 5

Conclusions and further work

The new requirements for a system able to retrieve images from very large databases based only on their visual content are motivating a lot of research on this topic. This paper addresses the problem by means of an algorithm based on logistic regression.

Since the user looks for images which are similar to his/her query, this defines a set whose indicator function, appropriately transformed by the logit mapping, is the output of the model to be fitted; its inputs are the low level image features directly extracted from the image. The main advantage of the method is the facility of incorporating the user's feedback. Its main drawback is the lack of sufficient information (too small a sample) to fit the model, since the number of inputs (image features) is usually high. This has been addressed by means of partial models that get the output from each subset of the inputs whose components are semantically related. The problem of combining the information from the different models, which is a data fusion problem, is addressed by using an ordered weighted averaging (OWA) operator.

An experiment of image retrieval from a large database (about 4700 images) has been designed and executed by 40 users of different ages and backgrounds. Due to the difficulty of evaluating subjective similarity in an objective way, the goal of the experiment was to retrieve a requested image. Results show that this could be done in an average of less than four (3.79) cycles of selection/ordering/presentation, for which the user selects an average of 3.5 positive and 5.7 negative samples per cycle. We consider this to be a good preliminary result that shows the usefulness of the proposed algorithm. As a side result, a cluster of the users based on their behaviour when faced with the system has been done. Results show four clear groups of users, depending on their personal attitude (patient/impatient) and on their ability to capture the visual resemblance between images.

As a further project, we intend to extend the model to categorical data, allowing the user to qualify the degree of similarity with several levels (probably, five) instead of just identifying the image as similar/dissimilar. This can be done with standard statistical techniques for categorical data regression. Also, the preprocessing of the inputs (low level image features) will be improved by using principal component or independent component analysis, which will improve the robustness and accuracy of the fitted models.

Another further project could be to apply this type of algorithms to music information retrieval (MIR) systems [24]. These kind of systems intend to retrieve musical information from large music databases using features such as pitch, timbre rhythm, or psicoacoustic measures extracted from music melodies. These features are part of the MPEG-7 standard [13].

Another interesting question could be to analyze the precision of the estimated relevance probabilities by using confidence intervals over the estimated values. This alternative would lead to another question: how to aggregate and order the information provided by intervals instead of single probability values.

Bibliography

[1] A. Agresti. *An Introduction to Categorical Data Analysis*. Wiley, 1996.

[2] A. Agresti. *Cualitative Data Analysis*. Wiley, second edition, 2002.

[3] G. Ciocca and R. Schettini. A relevance feedback mechanism for content-based image retrieval. *Information processing and management*, 35(1):605–632, 1999.

[4] P. Clough, H. Müller, and M. Sanderson1. *The CLEF 2004 Cross-Language Image Retrieval Track*. Springer Berlin / Heidelberg, second edition, 2005.

[5] I.J. Cox, M.L. Miller, T.P. Minka, T. V. Papathomas, and P.N. Yianilos. The bayesian image retrieval system, pichunter: Theory, implementation and psycholphysical experiments. *IEEE Transaction on image processing*, 9(1):20–37, 2000.

[6] E. de Ves, Domingo, J., G. Ayala, and P. Zuccarello. A novel bayesian framework for relevance feedback in image content-based retrieval systems. *Pattern Recognition*, 39:1622–1632, 2006.

[7] A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann, 1999.

[8] L. Duan, W. Gao, W. Zeng, and D. Zhao. Adaptive relevance feedback based on bayesian inference for image retrieval. *Signal Processing*, 85(2):395 – 399, February 2005.

[9] D. Filev and R. Yager. On the issue of obtaining owa operator weights. *Fuzzy Sets and Systems*, 94:157–169, 1998.

[10] R. Fuller and P. Majlender. On obtaining minimal variability owa operator weights. *Fuzzy Sets and Systems*, 136:203–215, 2003.

[11] T. Gevers and A. W. M. Smeulders. The pictoseek www image search system. In *IEEE International Conference on Multimedia Computing and Systems*, pages 264–269, Florence, Italy, July 1999.

[12] Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: Querying databases through multiple examples. In *Proc. 24th Int. Conf. Very Large Databases (VLDB)*, pages 264–269, New York, USA, 1999.

[13] A.T. Lindsay and J. Herre. Mpeg-7 and mpeg-7 audio - an overview. *Journal of the Audio Engineering Society*, 49:589–594, July/August 2001.

[14] M. O'Hagan. Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic. In *Proc. of 22nd Annu. IEEE Asilomar Conf. on Signals*, pages 681–689, Pacific Grove, CA., 1988.

[15] A. Pentland, R.W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254, 1996.

[16] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2006. ISBN 3-900051-07-0.

[17] J.J Rocchio. *Relevance Feedback in Information Retrieval. In SMART Retrieval System-Experiments in automatic Document Processing*. Prentice Hall, New York, 1971.

[18] Y. Rui, S. Huang, M. Ortega, and S. Mehrotra. Relevance feeback: a power tool for interactive content-based image retrieval. *IEEE Transaction on circuits and video technology*, 8(5), 1998.

[19] R.(ed.) Slowinski. *Fuzzy sets in decision analysis, operations research and statistics*. Kluwer Academic Publishers, 1998.

[20] A.W.M. Smeulders, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE transactions on Pattern Analysis and Machine Intellingence*, 22(12):1349–1379, 2000.

[21] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, Berlin, 2003.

[22] Z. Stejic, Y. Takama, and K. Hirota. Mathematical aggregation operators in image retrieval: effect on retrieval performance and role in relevance feedback. *Signal processing*, 85:1297–324, 2005.

[23] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

[24] Ch. Wan and M. Liu. Content-based audio retrieval with relevance feedback. *Pattern Recognition Letters*, 27:85–92, 2006.

[25] R.R. Yager. On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Trans. Systems Man Cybernet*, 18:183–190, 1988.

[26] X.S. Zhou and T.S. Huang. Unifying keywords and visual contents in image retrieval. *IEEE Multimedia*, 9(2):23–33, 2002.

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

Appendix A

International publications - Pattern Recognition Journal (Elsevier)

Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition 40 (2007) 2621 – 2632

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/pr

Applying logistic regression to relevance feedback in image retrieval systems

T. León[a], P. Zuccarello[b], G. Ayala[a,*], E. de Ves[b], J. Domingo[c]

[a]*Department of Statistics and Operations Research, University of Valencia, Spain*
[b]*Department of Informatics, University of Valencia, Avda. Vicente Andrés Estellés, 1. 46100 Burjasot, Spain*
[c]*Institute of Robotics, University of Valencia, Spain*

Received 14 July 2006; received in revised form 21 December 2006; accepted 5 February 2007

Abstract

This paper deals with the problem of image retrieval from large image databases. A particularly interesting problem is the retrieval of all images which are similar to one in the user's mind, taking into account his/her feedback which is expressed as positive or negative preferences for the images that the system progressively shows during the search. Here we present a novel algorithm for the incorporation of user preferences in an image retrieval system based exclusively on the visual content of the image, which is stored as a vector of low-level features. The algorithm considers the probability of an image belonging to the set of those sought by the user, and models the *logit* of this probability as the output of a generalized linear model whose inputs are the low-level image features. The image database is ranked by the output of the model and shown to the user, who selects a few positive and negative samples, repeating the process in an iterative way until he/she is satisfied. The problem of the small sample size with respect to the number of features is solved by adjusting several partial generalized linear models and combining their relevance probabilities by means of an ordered averaged weighted operator. Experiments were made with 40 users and they exhibited good performance in finding a target image (4 iterations on average) in a database of about 4700 images. The mean number of positive and negative examples is of 4 and 6 per iteration. A clustering of users into sets also shows consistent patterns of behavior.

Keywords: Visual information retrieval; Low-level image descriptors; Content-based image retrieval systems; Logistic regression

1. Introduction

The last few years have witnessed an increasing amount of pictorial information in different digital formats. Thus large image databases raise the need to retrieve relevant data efficiently. In this framework, content-based image retrieval (CBIR) systems are one of the most promising techniques for retrieving multimedia information [1–3]. CBIR systems are thought of as an improvement on traditional image retrieval systems based on textual information such as keywords. The new CBIR systems take advantage of valuable digital information held by the image itself. Visual features related to color, shape and texture are extracted in order to describe the image content [4]. The main drawback of textual image retrieval systems, that is, the annotator dependency, would be overcome in pure CBIR systems. Several papers have been published trying to integrate both approaches: textual and CBIR [5,6].

Image features are a key aspect of any CBIR system. A general classification can be made: low-level features (color, texture and shape) and high-level features (usually obtained by combining low-level features in a reasonably predefined model). High-level features have a strong dependency on the application domain, therefore they are not usually suitable for general purpose systems. This is the reason why one of the most important and developed research activities in this field has been the extraction of good low-level image descriptors. Obviously, there is an important gap between these features and human perception (a semantic gap). For this reason, different methods (mostly iterative procedures) have been proposed to deal with the semantic gap [7]. In most cases the idea underlying these methods is to integrate the information provided by the user into the decision process. This way, the user is in charge of guiding the search by indicating his/her preferences, desires

* Corresponding author. Tel.: +34 963543977; fax: +34 963544735.
E-mail addresses: Teresa.Leon@uv.es (T. León), pezuc@alumni.uv.es (P. Zuccarello), Guillermo.Ayala@uv.es (G. Ayala), Esther.deVes@uv.es (E. de Ves), Juan.Domingo@uv.es (J. Domingo).

0031-3203/$30.00
doi:10.1016/j.patcog.2007.02.002

and requirements to the system. The basic idea is rather simple: the system displays a set of images (resulting from a previous search); the user selects the images that are relevant (desired images) and rejects those which are not (images to avoid) according to his/her particular criterion; the system then learns from these training examples to achieve an improved performance in the next run. The process goes on iteratively until the user is satisfied.

The iterative algorithms which, in order to improve the set resulting from a query, require the user to enter his/her preferences in each iteration, are called relevance feedback algorithms [8]. These algorithms have been shown to provide a dramatic boost in retrieval system performance. Being part of this mainstream, this paper presents a new algorithm for relevance feedback in image databases based on logistic regression models.

A query can be seen as an expression of an information need to be satisfied. Any CBIR system aims at finding images relevant to a query and thus to the information need expressed by the query. The relationship between any image in the database and a particular query can be expressed by a relevance value. This relevance value relies on the user-perceived satisfaction of his/her information need. The relevance value can be interpreted as a mathematical probability (a relevance probability). The notion of relevance probability is not unique because different interpretations have been given by different authors. In this paper a relevance probability $\pi(I)$ is a quantity which reflects the estimation of the relevance of the image I with respect to the user's information needs. Initially, every image in the database is equally likely, but as more information on the user's preferences becomes available, the probability measure concentrates on a subset of the database. The iterative relevance feedback scheme proposed in the present paper is based on logistic regression analysis for ranking a set of images in decreasing order of their evaluated relevance probabilities.

Logistic regression is based on the construction of a linear model whose inputs, in our case, will be the image characteristics extracted from the image I and whose output is a function of $\pi(I)$. In logistic regression analysis, one of the key features to be established is the order of the model to be fitted. The order of logistic regression model, the number of image characteristics, and the number of relevant (positive/negative) images the user is prompted to select, are strongly related. The order of the model must be in accordance with the reasonable amount of feedback images requested from the user. For example, it is not reasonable for the user to select 40 images in each iteration; a feedback of 5/10 images would be acceptable. This requirement leads us to group the image features into n smaller subsets, each consisting of semantically related characteristics. The outcome of this strategy is that n smaller regression models must be adjusted: each sub-model will produce a different relevance probability $\pi_k(I)$ $(k = 1, \ldots, n)$. We then face to the question of how to combine the $\pi_k(I)$ in order to rank the database according to the user's preferences. We tackled this problem by making use of the so-called OWA (*ordered weighted averaging*) operators which were introduced by Yager [9] and provide a consistent and versatile way of aggregating multiple inputs into one single output.

Section 2 describes related work addressing issues of feature relevance computation. Section 3 presents and explains our approach in detail. Next, Section 3.1 describes the low-level features extracted from the images and used to retrieve them. After that, in Section 4 we present experimental results which evaluate the performance of our technique using real-world data. Finally, in Section 5 we extract conclusions and point to further work.

2. Related work

Relevance feedback is a term used to describe the actions performed by a user to interactively improve the results of a query by reformulating it. An initial query formulated by a user may not fully capture his/her wishes. This is due to several reasons: the complexity of formulating the query, lack of familiarity with the data collection procedures, or inadequacy of the available features. Users then typically change the query manually and re-execute the search until they are satisfied. By using relevance feedback, the system learns a new query that better captures the user's need for information.

In recent years, several methods have been developed to guide the searching process in a retrieval system. All these techniques can be roughly classified into two different groups:

- *Query point movement*: The method of the query point movement approach is to construct a new query point that is supposed to be close to the relevant results and far from those which are non-relevant. The best-known approach for achieving query point movement is based on a formula initially developed by Rocchio in the context of textual information retrieval [10].
- *Reshaping distance functions*: the objective of this approach is to modify the distance function in such a way that it can improve the query results according to the user's criterion.

A procedure belonging to the query point movement group was proposed by Ciocca and Schettini [11], who introduce a very simple algorithm for computing a new query point **Q** that can better represent the images of interest to the user. The procedure takes the set of relevant images the user has selected and computes a new point based on the standard deviation of the features used, computed separately one by one. Obviously, this ignores the dependency between image features, which is particularly important when they are values sampled from continuous functions.

Another implementation of point movement strategy consists of using the Bayesian methodology. Cox uses an adaptive Bayesian scheme which incorporates user preferences by means of a model of the user [12]. This model, together with the prior, gives rise inductively to a probability distribution on the event space. Experiments show that retrieval performance can be improved considerably by using such relevance feedback approaches. Relevance feedback has been also considered as a Bayesian classification problem by Duan et al. [13].

Yet another approach was taken by Rui et al., who propose an interactive retrieval approach which takes into account the user's high-level-query and perception subjectivity by dynamically updating certain weights [7]. Specifically, in this paper the images are represented by vectors of weights in the space of low-level characteristics; these weights capture the importance of components within a vector as well as their importance across different vectors over the entire data set. The system then uses relevance feedback to update queries so as to place more weight on relevant elements and less on irrelevant ones.

The system *MindReader* proposed by Ishigawa uses a method that combines ideas from the query-point movement and axis re-weighting [14]. The goal of this method is, given n images selected by a user in the relevance feedback step, to compute the coefficients of a distance function (namely, the distance matrix M) at the same time as the best query point q that represents the n images selected by the user in the relevance feedback procedure. By solving a minimization problem on the parameter estimation process, the authors conclude that the best M matrix (restricted to diagonal matrices only) is given by $m_{ij} \propto 1/\sigma_j^2$, σ_j being the standard deviation of the jth vector component.

Unlike the present paper, our previous work concerning relevance feedback CBIR algorithms was focused on a Bayesian strategy [15]. We followed the idea of modeling user preferences as a probability distribution. The person manifests his/her preferences on the set resulting from a query. The chosen images are considered as a sample. The previous information is modeled as the prior distribution and the choices are incorporated into the posterior distribution. The main advantage of our approach was to specifically work with a prior distribution such that the posterior distribution belongs to the same conjugate family of distributions.

3. A novel relevance feedback mechanism based on logistic regression

In this section, we will present the different elements used in our algorithm: image descriptors, how the relevance probabilities are calculated, the aggregation operators used to combine the different relevance probabilities and the global procedure.

Fig. 1 shows the different modules which integrate the proposed relevance feedback algorithm. The inputs and outputs of each module are also shown.

3.1. Visual features

This section deals with the low-level features the system uses for predicting human judgment of image similarity. Most of the features used are very simple, since our main goal is not to test the features as such, but only to use them as a tool to evaluate the new relevance feedback procedure. The relevance feedback methodology we have developed can be applied without changes to any image indexing and retrieval methods, even certainly a different set of features will give different results. With respect to the grouping of the features in smaller subsets, it is a resource to be able to apply logistic regression with a very small sample size. The results could change, too, by using a different grouping scheme.

Our system can currently work with different features that are obtained by preprocessing each image in the database. Amongst these characteristics we can mention those which represent chromatic information and those related to textures present in the image:

Color representation: The current version of the system incorporates as chromatic information of the image:
A histogram of the HS (Hue, Saturation) values of the image pixels: these values are obtained after conversion to HSV color space and quantization into $(H \times S)=(10\times 3)=30$ color bins. This 2D histogram is flattened by rows (H-component) giving a vector of 30 features.
Texture representation: The system currently works with information about textures in the image. This information is embodied as:
The granulometric cumulative distribution function. A granulometry is defined from the morphological opening of the texture using a convex and compact subset containing the origin as structuring element [16]. In our case we have

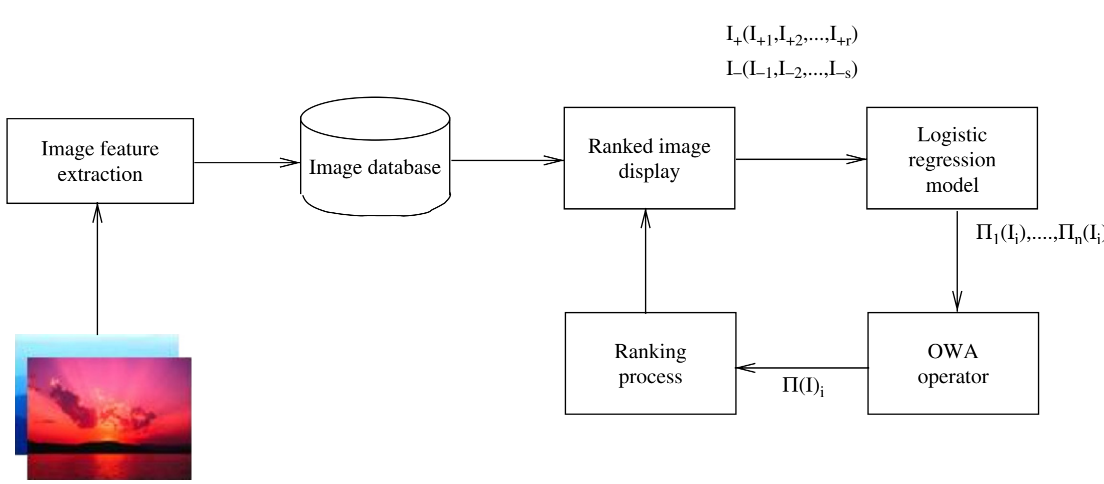


Fig. 1. Algorithm used.

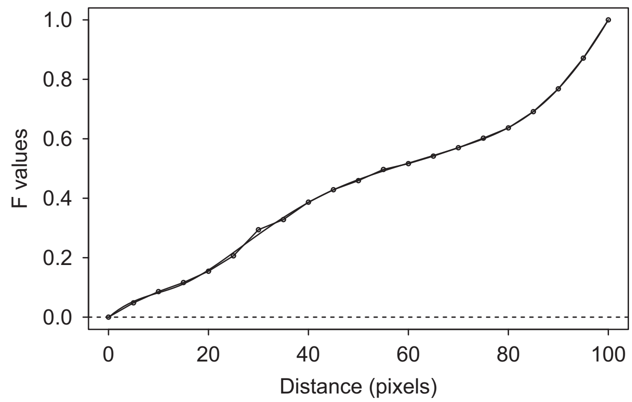


Fig. 2. Granulometry distribution function F together with its approximation with a 10 coefficient B-spline base.

used a horizontal and a vertical segment as the structuring elements.

In this paper we have approximated the cumulative granulometric function sampled from 0 to 100 pixels at intervals of 5 pixels by using a spline basis with 10 basis functions. Fourier basis is more appropriate for periodic functions, which is not the case here. Wavelets, on the other hand, are especially appropriate when the function presents intervals of high variation or even discontinuities together with relatively smooth areas. Since our functions are continuous, smooth and increasing, a spline basis can give a highly accurate representation; moreover, as splines are functions of local support, their coefficients are related with particular parts of the function support, which is appropriate in cases like this in which the value of the granulometry is directly related to the size of the structuring element, which is in turn related to the size of the particles in the texture of the image. Fig. 2 shows a typical granulometry distribution function F sampled from 0 to 100 pixels at intervals of 5 pixels (with circles) together with its approximation with a 10 coefficient B-spline basis f_b (smooth line). As can be appreciated visually, the fit is very good; numerically, the error can be calculated as

$$Error = \frac{1}{N}\sqrt{\sum_{i=1}^{N} \mid F(x_i) - F_b(x_i)|^2}.$$

N being the number of points and F_b the B-spline approximation of F; error is 2.6×10^{-5} for this case; as a comparison, the mean value of F is 0.438.

3.2. Logistic regression and relevance probabilities

At each iteration, a sample is evaluated by the user selecting two sets of images: the examples or positive images and the counter-examples or negative images. Let us consider the (random) variable Y giving the user evaluation where $Y = 1$ means that the image is positively evaluated and $Y = 0$ means a negative evaluation.

Each image in the database has been previously described by using low-level features in such a way that the j-th image has the k-dimensional feature vector x_j associated. Our data will consist of (x_j, y_j), with $j = 1, \ldots, n$ where n is the total number of images, x_j is the feature vector and y_j the user evaluation (1 = positive and 0 = negative). The image feature vector x is known for any image and we intend to predict the associated value of Y. The natural framework for this problem is the generalized linear model (GLM). In this paper, we have used a logistic regression where $P(Y = 1 \mid x)$ i.e. the probability that $Y = 1$ (the user evaluates the image positively) given the feature vector x, is related with the systematic part of the model (a linear combination of the feature vector) by means of the logit function. GLMs extend ordinary regression models to encompass non-normal response distributions and modeling functions of the mean. Most statistical software has the facility to fit GLMs. Logistic regression is the most important model for categorical response data. Logistic regression models are also called *logit* models. They have been successfully used in many different areas including business applications and genetics. For a binary response variable Y and p explanatory variables $X_1, \ldots, X_p$, the model for $\pi(x) = P(Y = 1 \mid x)$ at values $x = (x_1, \ldots, x_p)$ of predictors is

$$logit[\pi(x)] = \alpha + \beta_1 x_1 + \cdots + \beta_p x_p, \tag{1}$$

where $logit[\pi(x)] = \ln(\pi(x)/(1 - \pi(x)))$. The model can also be stated directly specifying $\pi(x)$ as

$$\pi(x) = \frac{\exp(\alpha + \beta_1 x_1 + \cdots + \beta_p x_p)}{1 + \exp(\alpha + \beta_1 x_1 + \cdots + \beta_p x_p)}. \tag{2}$$

The parameter β_i refers to the effect of x_i on the log odds that $Y = 1$, controlling the other x_j. The model parameters are obtained by maximizing the *likelihood function* given by

$$l(\beta) = \prod^{n} \pi(x_i)^{y_i}[1 - \pi(x_i)]^{1-y_i}. \tag{3}$$

The maximum likelihood estimators (MLE) of the parameter vector β are calculated by using an iterative method. A detailed explanation can be found in Ref. [17, p. 192]. We have used the public domain statistical program R [18]. In particular, the function *glm* from the package *MASS* [19]. An additional advantage of using generalized linear models is that, if each feature has a known semantic meaning, the value of the parameter associated to it has to do with its importance for the viewer, and the width of its confidence interval might be considered to be related with the certainty with which the viewer uses that feature. Nevertheless, in our case the semantic meaning is not clearly associated to a particular parameter, but to a group of them (color parameters, texture parameters, etc.) so these considerations are barely useful.

In the first steps of the procedure, we have a major difficulty when having to adjust a global regression model in which we take the whole set of variables into account, because the number of images (the number of positive plus negative images chosen by the user) is typically smaller than the number of characteristics. In this case, the regression model adjusted has as many parameters as the number of data and many relevant variables could be not considered. On the other hand it is not realistic

Table 1
Illustrating examples of OWA aggregation values

W	$f(a_1, \ldots, a_n)$
$(1, 0, \ldots, 0)$	$\max_i a_i$
$(0, 0, \ldots, 1)$	$\min_i a_i$
$(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})$	$\frac{1}{n}\sum_{j=i}^{n} a_i$

to ask the user to make a great number of positive and negative selections from the very beginning; therefore we think that the difficulty cannot be avoided in this way. In order to solve this problem, our proposal is to adjust different smaller regression models: each model considers only a subset of variables consisting of semantically related characteristics of the image. Consequently, each sub-model will associate a different relevance probability to a given image x, and we face the question of how to combine them in order to rank the database according to the user's preferences. We can see this question as an information fusion problem.

3.3. Aggregating the relevance probabilities

Let us denote as $\pi_1(x), \pi_2(x), \ldots, \pi_n(x)$ the different relevance probabilities associated with a given image x. Each one of them has been obtained separately by using different regression models and we need to associate a final probability $\pi(x)$ by aggregating the information provided by each $\pi_j(x)$, $(j = 1, \ldots, n)$. Mathematical aggregation operators transform a finite number of inputs into a single output and play an important role in image retrieval. In Ref. [20] the authors compare the effect of 67 operators applied to the problem of computing the overall image similarity, given a collection of individual feature similarities. Their results show how important for retrieval performance the choice of the aggregation operator is. A comprehensive overview, as well as the classification, of mathematical aggregation operators can be found in Ref. [21]. We have not used any of the 67 operators reviewed; instead, we decided to use the so-called OWA operators to aggregate our relevance probabilities. These operators were introduced in Ref. [9]. Since then they have been successfully applied in different areas: decision making, expert systems, neural networks, fuzzy systems and control, etc. An OWA operator of dimension n is a mapping $f : \mathbb{R}^n \to \mathbb{R}$ with an associated weighting vector $W = (w_1, \ldots, w_n)$ such that $\sum_{j=1}^{n} w_j = 1$ and where $f(a_1, \ldots, a_n) = \sum_{j=1}^{n} w_j b_j$ where b_j is the j-th largest element of the collection of aggregated objects $a_1, \ldots, a_n$. The particular cases shown in Table 1 can better illustrate the idea underlying OWA operators.

Note that no weight is associated with any particular input; instead, the relative magnitude of the input decides which weight corresponds to each input. In our application, the inputs are relevance probabilities and this property is very interesting because we do not know, a priori, which set of visual descriptors will provide us with the *best* information.

As OWA operators are bounded by the max and min operators, Yager introduced a measure called *orness* to characterize the degree to which the aggregation is like an *or* (max) operation:

$$\text{orness}(W) = \frac{1}{n-1}\sum_{i=1}^{n}(n-i)w_i. \tag{4}$$

Note that for $W = (1, 0, \ldots, 0)$, $\text{orness}(W) = 1$, for $W = (0, 0, \ldots, 1)$, $\text{orness}(W) = 0$ and for $W = (1/n, 1/n, \ldots, 1/n)$, $\text{orness}(W) = 0.5$. This author also introduced the concept of *dispersion* or *entropy* associated with a weighting vector:

$$Disp(W) = \sum_{i=1}^{n} w_i \ln w_i. \tag{5}$$

$Disp(W)$ tries to reflect how much of the information in the arguments is used during an aggregation based on W.

Clearly, the vector of weights W can be pre-fixed, but a number of approaches have also been suggested for determining it according to different criteria. For instance, in Ref. [22] an algorithm which can be used to learn the weights from an observation of performance by others. One of the first methods developed was proposed by O'Hagan [23]. It provides us with the vector of weights for a given level of orness (optimism) which maximizes their entropy:

$$W = \operatorname{argmax} \sum_{i=1}^{n} w_i \ln w_i, \quad \text{subject to} \begin{cases} \alpha = \frac{1}{n-1}\sum_{i=1}^{n}(n-i)w_i, \\ \sum_{i=1}^{n} w_i = 1, \quad w_i \in [0, 1]. \end{cases} \tag{6}$$

Roughly speaking, when the maximum entropy vector of weights for a prefixed orness is calculated, the set of weights most similar to a discrete uniform distribution is obtained, in the sense that each weight tends to achieve its maximum value while the sequence of weights increases or decreases (depending on the orness value). By this we mean that if $W = (w_1, \ldots, w_n)$, then $w_1 \geqslant w_2 \geqslant \cdots \geqslant w_n$, or $w_1 \leqslant w_2 \leqslant \cdots \leqslant w_n$. This problem is not computationally easy to solve. Fuller and Majlender [24] have obtained the analytical expression of the maximum entropy weights. Their practical procedure involves the solution of a polynomial equation; however, the unique root of a rather flat polynomial is numerically difficult to obtain.

We have used a simpler procedure to generate the set of weights $W = (w_1, \ldots, w_n)$. They are obtained as a mixture of the binomial $Bi(n-1, p)$ and the discrete uniform probability distributions. A principal advantage of this choice is its flexibility and simplicity: the weights are easily obtained and are also easy to interpret; however, the main reason for choosing this procedure is that our practical experiments have shown that it works well for our case.

Notice that OWA operators with many of the weights close to their highest values will be *or-like* operators ($\text{orness(W)} \geqslant 0.5$), while those operators with most of the weights close to their lowest values will be *and-like* operators ($\text{orness(W)} \leqslant 0.5$). The sequence of weights that we can obtain as a mixture of a binomial and a discrete uniform distribution is not necessarily

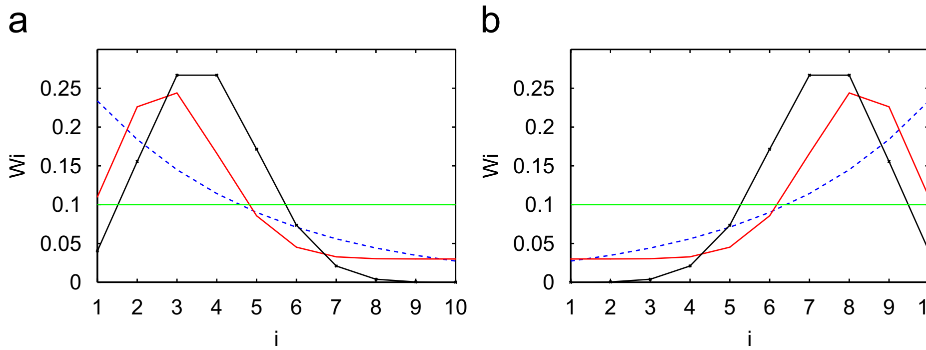


Fig. 3. Aggregation weights with $n=10$ obtained using the Fuller and Majlender method (blue dashed line) with orness α, probabilities of a $Bi(n-1, 1-\alpha)$ distribution (x-marked line), mixture with parameter λ of a $Bi(n-1, p)$ distribution and a discrete uniform (dotted line), and discrete uniform (continuous green line): (a) $\alpha=0.7$, $\lambda=0.7$ and $p=0.2143$, (b) $\alpha=0.3$, $\lambda=0.7$ and $p=0.7857$.

increasing or decreasing and the combination of the values for the parameter p of the binomial probability distribution and the mixture parameter λ allows us to obtain a parametric family of weights easily.

3.3.1. Some properties

Proposition 1. *Let $W=(w_1,\ldots,w_n)$ and $W'=(w'_1,\ldots,w'_n)$ be two vectors of weights such that orness $(W)=\alpha$ and orness $(W')=\beta$ with $\alpha,\beta\in[0,1]$, then $\forall\lambda\in[0,1]$ we have the orness of $(\lambda W+(1-\lambda)W')=\lambda\alpha+(1-\lambda)\beta$, where $\lambda W+(1-\lambda)W'=(\lambda w_1+(1-\lambda)w'_1,\ldots,\lambda w_n+(1-\lambda)w'_n)$.*

The proof follows from the definition of orness.

Proposition 2. *Let $W=(w_1,\ldots,w_n)$ be a vector of weights such that $w_i=\pi_{i-1}=\binom{n-1}{i-1}(1-\alpha)^{i-1}\alpha^{n-i}$, then orness$(W)=\alpha$.*

Proof. Note that the components of the vector of weights are the n probabilities of a binomial probability distribution with parameters $n-1$ and $p=1-\alpha$, therefore,

$$\begin{aligned}\text{orness}(W)&=\frac{1}{n-1}\sum_{i=1}^{n}(n-i)\pi_{i-1}\\&=\frac{1}{n-1}\left(n-\sum_{j=0}^{n-1}(j+1)\pi_j\right)\\&=\frac{1}{n-1}(n-(n-1)(1-\alpha)-1)\\&=\alpha.\end{aligned}\tag{7}$$

We can use these results to obtain different sets of weights for a given orness.

Corollary 1. *Let $W=(w_1,\ldots,w_n)=(\lambda\pi_0+(1-\lambda)1/n,\ldots,\lambda\pi_{n-1}+(1-\lambda)(1/n))$, where $\pi_j=P(X=j)$ with $X\sim B(n-1,p)$, then orness$(W)=\lambda(1-p)+(1-\lambda)0.5$.*

A direct consequence of this corollary is that, for a given orness α, we can construct a vector of weights as a mixture of binomial and discrete uniform probabilities. The relationship between α, λ, and p can be expressed as:

$$2\alpha-1=\lambda(1-2p),\tag{8}$$

i.e. the set of pairs $\{(p,\lambda): 2\alpha-1=\lambda(1-2p),\ \lambda\in(0,1]\}$ characterizes the different mixtures which give rise to the different vectors of weights for a given orness α.

We can distinguish three cases:

(1) If $\alpha<0.5\ \rightarrow\ 2\alpha-1<0$ (and $\lambda>0$) $\rightarrow\ 1-2p<0\ \rightarrow\ p>0.5$. Moreover, as $\lambda\leqslant 1$, we have that $1-2\alpha\leqslant 2p-1$ and then $p\geqslant 1-\alpha$ with $\lambda\in(0,1]$.
(2) Analogously, if $\alpha>0.5$ we obtain $p\leqslant 1-\alpha$ with $\lambda\in(0,1]$.
(3) Finally, if $\alpha=0.5$, then $\lambda=0$ and $p\in(0,1)$ or $p=0.5$ and $\lambda\in(0,1]$.

Fig. 3 illustrates a comparison of the aggregation of weights for $n=10$ obtained with the above-mentioned methods: Fuller and Majlender, a binomial probability density function, and a mixture of a binomial probability density function and a discrete uniform. In Fig. 3(a) we can see that the maximum entropy vector of weights might attach too great an importance to the biggest input, and also that the binomial probability distribution is “too extreme”, although, the addition of the uniform part of the mixture “smoothes” its shape.

3.4. General procedure

So far we have made a detailed description of the different theoretical components of our novel search strategy. It is now time to explain how we combine them into an efficient relevance feedback algorithm.

Let us assume a collection of images (the database) where a set of image features has been computed off-line for each image in the collection. Our particular choice of features has been: first, the 30 values of the HS-histogram bins; then, the granulometric cumulative distribution function is calculated for two different structuring elements: a vertical and a horizontal line, and sampled at intervals of 5 pixels between 0 and 100. This gives 20 values for each structuring element. Instead of using these values as raw data we approximate the function by expressing it in a B-spline basis of 10 basis functions and taking the coefficients as features. This add up to 10 coefficients per structuring element, so we have a total of 20, which added to the 30 values of the histogram results in a final vector of 50 features per image. See detailed explanation in Section 3.1. With respect to the grouping of the characteristics which are semantically related to apply the approach based on several sub-models, we have considered 10 groups, each one made by five consecutive characteristics. Note that six of these groups are related to color values and four to texture values. No group contains both types of characteristics. In Section 4 we discuss through an experiment the influence of having more or less groups, with the condition that each group contains only one type of feature, either color or texture. The choice of these particular groups has been motivated by common sense since it seems natural to group together color features, and in a different group from texture ones. Nevertheless, the choice of appropriate groups is an important and interesting topic because of two reasons: first, it may improve the performance of the retrieval algorithms and second, it may make apparent hidden groups that are into the viewer's mind (therefore, properly semantic) which he/she is unable to express in words or even realize of their existence. This is a possible line of research for future work or to be considered by people involved in artificial intelligence, concretely knowledge elicitation.

Let us also assume that the images are initially randomly ranked. Each iteration of the relevance feedback algorithm changes the ranking of the images according to a given set of data. By data we mean a user selection of positive and negative relevant images, and a set of aggregation weights.

An schematic description of the procedure is as follows:

Initialization: Images are randomly ranked.

Input parameters: Positive and negative relevant images are selected from amongst the whole collection. Let $\mathbf{I_P}$ be the set of positive samples, and $\mathbf{I_N}$ the set of negative samples. Let $W = (w_1, \ldots, w_n)$ also be the set of aggregation weights, where n is the number of relevance probabilities (outputs of the different logistic regression models) to be combined.

Logistic regression model: Using inputs selected in the previous step, several logistic regression models are fitted. Such models are applied to each image I_j in the database, obtaining their respective relevance probabilities, $(\pi_1(I_j), \ldots, \pi_n(I_j))$.

Aggregation and ranking: In order to obtain a unique relevance value, the relevance probabilities $\pi_1(I_j) \ldots \pi_n(I_j)$ should now be aggregated using the previously selected weights W (see Section 3.3 for a detailed description of OWA aggregation operator). Images are ranked according to the computed relevance values.

The numbers of positive and negative relevant images are not required to be equal. In the first iterations it is the usual case not to find many positive relevant images, therefore the number of images in set $\mathbf{I_P}$ is not very large. This is not the case for negative relevant selection, therefore the set $\mathbf{I_N}$ usually has many more images than $\mathbf{I_P}$.

When a user rejects an image by selecting it as negative, we assume that the user's wish for that particular image will not change at any point in the searching process. Therefore we have implemented a memory algorithm for the selection of negative relevant images. Negative selections are remembered through all iterations. In iteration r, the set of negative relevant images used as input for the logistic regression model, $\mathbf{I_N}$, is obtained as

$$\mathbf{I_N} = \mathbf{I_N^r} \cup \mathbf{I_N^{prev}}, \tag{9}$$

where $\mathbf{I_N^r}$ is the set of negative images selected by the user in iteration r, and $\mathbf{I_N^{prev}}$ is a subset of randomly selected images from user negative selections made in iterations 1 to $r-1$. The probability of a certain image, I_i, belonging to $\mathbf{I_N^{prev}}$ is

$$P(I_i) = \frac{i}{\sum_{q=1}^{r-1} q N_q}, \tag{10}$$

where I_i is each image selected as negative in iteration i, N_q is the number of negative selected images in iteration q, and r is the present iteration.

No memory was implemented for positive relevant selection; in each iteration the logistic regression models are fitted with just the positive choice of the current iteration. The rationale behind this is to allow the user to focus his/her search on progressively narrower sets and allow him/her to choose as negative samples images that were formerly positive, possibly because none of the images shown in former iterations were sufficiently similar to the target.

With respect to the choice of the orness value for the OWA aggregation operator, the user can choose it freely. He/she receives a previous explanation about its meaning: the idea is to start with a high orness value (allowing recalls of images similar only by one of the groups, which from his/her point of view means loosely similar) and decreasing later to a more restrictive choice (similarity by all of the groups) as the search proceeds and is more centered on really similar images. In most of the cases it was sufficient to make one change of the orness value during the search.

4. Experimental results

4.1. Experimental setup

The main objective of our algorithm is to find an image which is similar to what the user may have in mind. Therefore, the first step in the design of the experiments would be to define what is understood by "similar". Unfortunately, this is not easy since it depends on the user, and the goal of the algorithm is precisely to capture that notion of similarity that each user has, which can also change between different queries. Consequently, the valid criterion of similarity appears to be the user's opinion. This would have introduced an external variable into the experiment that would have masked the main goal: an objective evaluation of the system as such. That is why we have chosen to use an approach in which a given image has to be found. The search is considered successful if the image is ranked within the first 16. This number is arbitrary but we have checked that 16

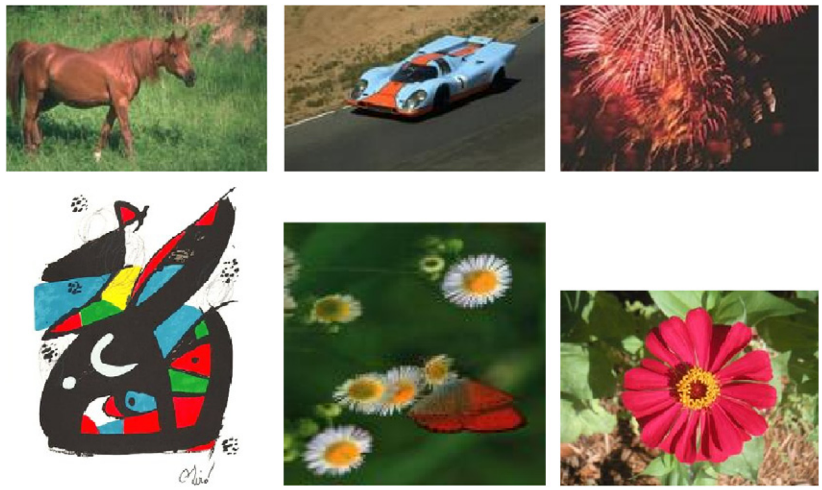


Fig. 4. Target images used in experiments.

images shown side by side is a reasonable number to localize a particular one at a first sight.

Once the criterion for termination has been adopted, the experiment will be designed by showing several images to the user; a choice of six images (the same for all users) was selected from a database of about 4700. The pictorial database was assembled using some images obtained from the web and others chosen by the authors. These images are classified as belonging to different themes such as flowers, horses, paintings, skies, textures, ceramic tiles, buildings, clouds, trees, etc. even though the category is not used at all during the search. The six target images are, in our experience, representative of different types and levels of difficulty. They are displayed in Fig. 4.

For each target image the search proceeds iteratively. In each iteration the user has to select some relevant images (similar to the target according to his/her judgment) and others significantly different from the target. The number of images of each type is left to the user, although two conditions must be fulfilled: at least one relevant and one irrelevant images must be selected and the total number of selections has to be greater than the maximum number of features jointly considered. For instance, if we group the features in groups of five then the minimum number of images evaluated has to be five per iteration.

The algorithm proceeds as explained in previous sections and shows the images in the database ordered according to their similarity to the target, in groups (screens) of 32 (the 16 more similar and the 16 more dissimilar). If the target appears in the first 16, it is considered to have been found; otherwise the user can move backwards or forwards to see more images in rank order and a new iteration of choosing/search/showing begins.

All the experiments have been performed using a graphical user interface GVI (see Fig. 5), a system previously programmed by our research group that allows the easy introduction of new images, new features associated to each image and new similarity measures, being a powerful and flexible tool. GVI is oriented to an expert user with some basic knowledge of image processing and content-based retrieval. The creation of retrieval systems for multimedia data is a non-trivial problem, which could be facilitated with a system like this.

The variables we intend to analyze in our experiment are:

- The number of iterations used by each person to find the target. This is the most important variable for a system of this type.
- The number of iterations used to find an image, independently of the user. This is intended to evaluate if the nature of the image has an important influence on the result.
- The number of relevant and non-relevant images chosen as a function of the image.
- The same as a function of the iteration. This should be related to the progressive convergence towards the target.
- The position of the target in the rank for each iteration.

All these variables were evaluated for 40 users who did the test for the six chosen images; amongst them there were computer programmers, mathematicians, children and one graphical designer, all of different ages and levels of exposure to computers.

A general graphical and numerical description is provided in Section 4.2. Then, a non-supervised classification of the user behavior will be given in Section 4.3.

4.2. Descriptives

Firstly, a descriptive study of the previously indicated variables will be given. For each image and trial we consider the number of iterations, the mean rank of the target image and the mean number of positively and negatively chosen images. These means are calculated throughout the iterations of each trial. The description will be given in Table 3 in terms of the first, second and third quartiles in the column under the heading "All". It is notable that the number of iterations goes from 3.12 to 4.43, which can be considered to be quite small. We have not taken into account the time for performing the experiment

Fig. 5. Graphical user interface used in the experiments.

because it depends on the degree of knowledge of the application that the user has. However, a typical trial is usually done by looking at three to four displays with a duration of 10–20 s per display, which adds up to a total time of around 1 min per trial.

Concerning the mean rank, it should be noticed that in our experimental setup, the target image is randomly located in the last third of the database, i.e. its initial rank is greater than 3000. During the iterations the rank decreases until finishing before 16th; in this process the quartiles of the rank go from 195 to 343, with a median of 274.

Regarding the number of evaluated images, as usual in these kind of procedures, the user does not select a great number, but consistently selects a greater number of negative samples (median of 8.22) than positive ones (median of 3.89).

Clearly, all these variables depend heavily on the behavior of each user. This will be carefully analyzed in the next subsection.

The dependence on the image will be evaluated by means of a multiple box plot displayed in Fig. 8. Two apparent facts have to be pointed out. The medians are very similar, except for image fue21 (a photograph of fireworks) which is the easiest one. However, the variability is clearly different between images. The largest box corresponds to image co25 (a car), but there are outliers in most of the data groups, suggesting that certain images are particularly difficult for certain users. Again, no common behavior appears to arise amongst all the users, which leads us to try a non-supervised classification of users.

4.3. Non-supervised classification

We will describe the trial corresponding to a given user by means of four variables: the mean number of iterations, the mean ranking of the image that we are looking for throughout the iterations, the mean number of images positively evaluated and the mean number of images negatively evaluated i.e. the experiment for a user is summarized as a four feature vector. The results observed in a trial are the combined consequence of the software tool to be evaluated and the characteristics of the particular user. It seems to us very interesting to find groups of similar users by taking into account the feature vector which has just been mentioned. Therefore, we have computed an agglomerative hierarchical clustering of the data set to explore its structure. Fig. 6. shows the corresponding dendrogram. At first sight, four different clusters are observed. In order to confirm this, we have also used a non-hierarchical partitioning method.

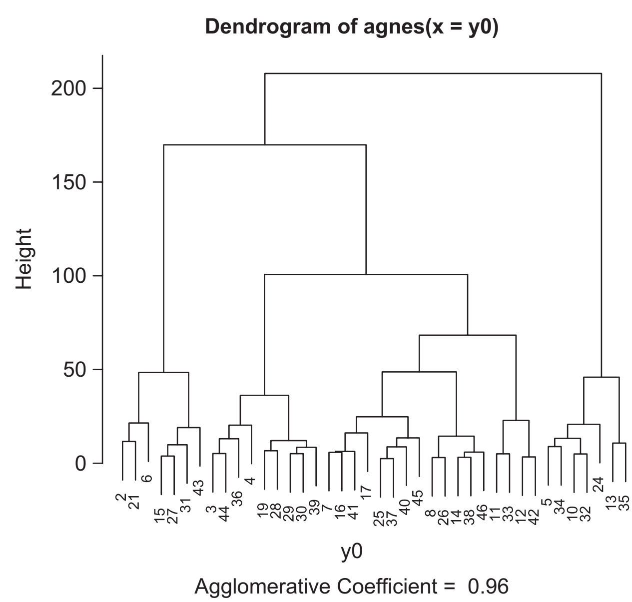


Fig. 6. Dendrogram.

Partitioning methods require that the number of clusters, k, be given by the user; the clustering method which seems natural in this context is the partitioning around medoids. The first step is to find k representative objects or medoids from among the observations in the data set. These observations should represent the structure of the data. The k clusters are constructed by assigning each observation to the nearest medoid. We have used the silhouette widths for assessing the best number of clusters.

Let us briefly recall the definitions of silhouette and silhouette plot in a graphical display, in which for each observation i, a bar is drawn, representing its silhouette width.

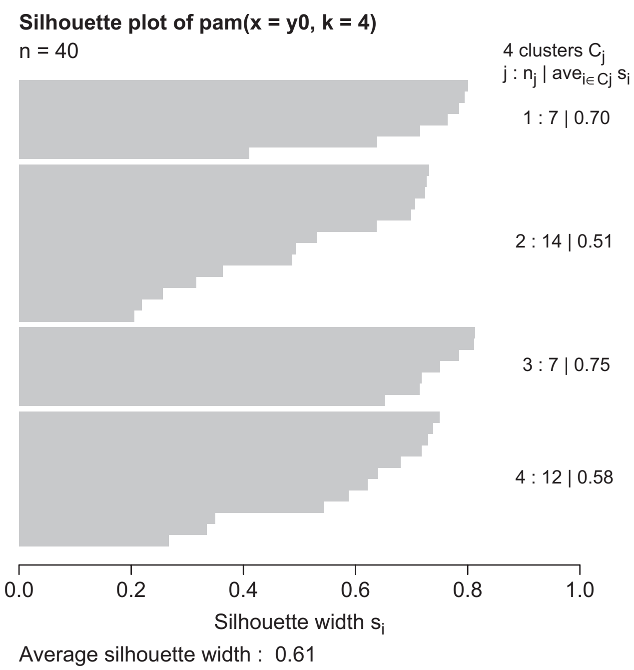


Fig. 7. Silhouette observed using four groups.

For each observation, i, the silhouette width, $s(i)$, is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}, \tag{11}$$

where $a(i)$ is the average dissimilarity between i and all other points of the cluster to which i belongs, and $b(i)$ is the dissimilarity between i and its neighboring cluster i.e. the nearest one to which i does not belong. Clearly, observations with a large $s(i)$ are very well clustered, $s(i)$ around 0 means that the observation is between two clusters, and observations with a negative $s(i)$ are probably placed in the wrong cluster.

Moreover, the average silhouette allows us to select the best number of clusters: a clustering can be performed for several values of k, the number of clusters and then we choose the value of k with the largest overall average silhouette width. This value is equal to four in our case. The silhouette widths, together with the average silhouette width for each group and the (global) average silhouette width, can be found in Fig. 7. Notice that the average silhouette width is equal to 0.61, suggesting that the structure of the data set is quite well defined.

We have also performed a k-means analysis with $k = 4$ and found exactly the same clustering, this fact confirming the robustness of our results.

We would like to mention that these four clusters are determined both for the personality of the user and also the kind of experience that we have carried out. We are aware that our experiment is not the ideal one for testing our application but, in our opinion, it is not easy to improve on. The aim of our procedure is not to help the user retrieve a particular image in the data set, as in the experiment. Instead, its aim is to help users find those images with the features that they have in mind.

Table 2 displays the medoids or representative individuals of the different clusters and some descriptive statistics for the four clusters. This table allows us to interpret the particular characteristics of the individuals in each cluster.

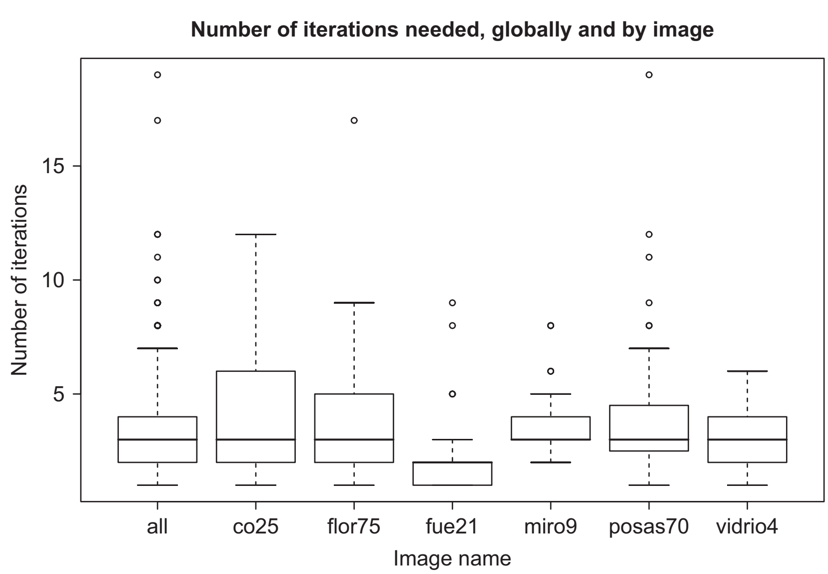


Fig. 8. Box plots showing the number of iterations for each image

Table 2
Medoids

Medoid	Iterations	Rank	Positive	Negative
1	3.86	459.56	3.48	8.22
2	4.70	314.08	3.13	12.17
3	3.00	103.24	7.45	10.44
4	5.57	224.41	5.67	9.30

Table 3
Basic descriptives of the variables mean number of iterations, mean rank, mean of positively and negatively evaluated images

	Group 1	Group 2	Group 3	Group 4	All
Iterations					
1st Qu.	4.14	3.33	2.14	3.40	3.12
Median	4.43	3.79	2.29	4.17	3.79
3rd Qu.	4.86	3.74	3.40	4.43	4.43
Ranks					
1st Qu.	419.2	288.6	85.6	195.3	195.3
Median	448.9	318.0	103.2	222.0	274.0
3rd Qu.	460.9	337.3	114.6	238.8	343.9
Positive					
1st Qu.	3.30	3.18	3.29	3.65	3.29
Median	3.48	3.77	4.79	5.12	3.89
3rd Qu.	4.62	4.14	7.13	6.57	3.90
Negative					
1st Qu.	5.27	5.80	7.20	6.65	5.74
Median	5.70	8.46	8.97	8.09	8.22
3rd Qu.	8.35	10.99	10.45	9.39	10.06

The users belonging to cluster 1 are the impatient ones: they choose just a few positive and a few negative images and, although they find their objective, they do it in the end or with ups and downs (large average position). On the other hand, the individuals in cluster 3 are the most patient: they find the target with the minimum number of iterations and the best average position; these users make a lot of positive selections and also many negative selections. The individuals clustered in 2 mark many images negatively but they only select a few positive images, their average position through the search is smaller than the average position for cluster 1. The users classified as cluster number 4 need more iterations to achieve their objective, and although they make many positive selections they are not very lucky in their choices. (Table 3).

In any case, we think that we have obtained quite good results considering that the average number of iterations in every cluster is very reasonable.

4.4. *Influence of the grouping of characteristics*

A potentially important point is the influence on the results of the way in which the components of the feature vector are grouped in sub-models to be fitted by the regression algorithm and it is reasonable to pose the question of what is better: few sub-models with a relatively large number of characteristics or more sub-models with a few components each. Apart from avoiding the mixing of features of different nature (color with texture, in our case) there is no a clear answer. An experiment has been done to en-light this issue by using groups of 3, 5 and 7 features to recall seven different images, being the other parameters as in former experiment, and equal for the three trials. Results in terms of number of trials to find the target can be seen in Table 4.

Table 4
Number of iterations to recall an image depending on the size of the chosen sub-models

Sub model size	Image number							Mean	St. dev.
	1	2	3	4	5	6	7		
3	4	10	1	1	6	2	6	4.28	3.30
5	4	2	1	7	4	1	5	3.43	3.30
7	1	3	2	5	6	4	8	4.14	2.41

The mean and standard deviation of the number of iterations appears to be slightly smaller for models of five characteristics than for the others, although given the sample size (7 images) this cannot be considered significant. This experiment has suggested us the convenience of choosing the five characteristics model, taking into account essentially the balance between the robustness of the model (in principle, bigger for larger groups) and the minimum number of images a user is asked to select.

5. Conclusions and further work

The new requirements for a system able to retrieve images from very large databases based only on their visual content are motivating a lot of research on this topic. This paper addresses the problem by means of an algorithm based on logistic regression.

Since the user looks for images which are similar to his/her query, this defines a set whose indicator function, appropriately transformed by the *logit* mapping, is the output of the model to be fitted; its inputs are the low-level image features directly extracted from the image. The main advantage of the method is the facility of incorporating the user's feedback. Its main drawback is the lack of sufficient information (too small a sample) to fit the model, since the number of inputs (image features) is usually high. This has been addressed by means of partial models that get the output from each subset of the inputs whose components are semantically related. The problem of combining the information from the different models, which is a data fusion problem, is addressed by using an ordered weighted averaging (OWA) operator.

An experiment of image retrieval from a large database (about 4700 images) has been designed and executed by 40 users of different ages and backgrounds. Due to the difficulty of evaluating subjective similarity in an objective way, the goal of the experiment was to retrieve a requested image. Results show that this could be done in an average of less than four (3.79) cycles of selection/ordering/presentation, for which the user selects an average of 3.5 positive and 5.7 negative samples per cycle. We consider this to be a good preliminary result that shows the usefulness of the proposed algorithm. As a side result, a cluster of the users based on their behavior when faced with the system has been done. Results show four clear groups of users, depending on their personal attitude (patient/impatient) and on their ability to capture the visual resemblance between images.

As a further project, we intend to extend the model to ordinal data, allowing the user to qualify the degree of similarity with several levels (probably, five) instead of just identifying the image as similar/dissimilar. This can be done with standard statistical techniques for categorical data regression. Also, the preprocessing of the inputs (low-level image features) will be improved by using principal component or independent component analysis, which will improve the robustness and accuracy of the fitted models.

Acknowledgements

This paper has been supported by grant TIN2006-10134.

References

[1] A.W.M. Smeulders, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Ttrans. Pattern Anal. Mach. Intell. 22 (12) (2000) 1349–1379.
[2] A. Pentland, R.W. Picard, S. Sclaroff, Photobook: content-based manipulation of image databases, Int. J. Comput. Vision 18 (3) (1996) 233–254.
[3] T. Gevers, A.W.M. Smeulders, The pictoseek www image search system, in: IEEE International Conference on Multimedia Computing and Systems, Florence, Italy, July 1999, pp. 264–269.
[4] A. Del Bimbo, Visual Information Retrieval, Morgan Kaufmann, Los Altos, CA, 1999.
[5] P. Clough, H. Müller, M. Sanderson, The CLEF 2004 Cross-Language Image Retrieval Track, second ed., Springer, Berlin, Heidelberg, 2005.
[6] X.S. Zhou, T.S. Huang, Unifying keywords and visual contents in image retrieval, IEEE Multimedia 9 (2) (2002) 23–33.
[7] Y. Rui, S. Huang, M. Ortega, S. Mehrotra, Relevance feedback: a power tool for interactive content-based image retrieval, IEEE Trans. Circuits Video Technol. 8 (5) (1998).
[8] X.S. Zhou, T.S. Huang, Relevance feedback for image retrieval: a comprehensive review, Multimedia Systems 8 (6) (2003) 536–544.
[9] R.R. Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making, IEEE Trans. Systems Man Cybernet. 18 (1988) 183–190.
[10] J.J. Rocchio, Relevance feedback in information retrieval. In SMART Retrieval System-Experiments in automatic Document Processing, Prentice-Hall, New York, 1971.
[11] G. Ciocca, R. Schettini, A relevance feedback mechanism for content-based image retrieval, Inf. Process. Manage. 35 (1) (1999) 605–632.
[12] I.J. Cox, M.L. Miller, T.P. Minka, T.V. Papathomas, P.N. Yianilos, The bayesian image retrieval system, pichunter: theory, implementation and psychophysical experiments, IEEE Trans. Image Process. 9 (1) (2000) 20–37.
[13] L. Duan, W. Gao, W. Zeng, D. Zhao, Adaptive relevance feedback based on bayesian inference for image retrieval, Signal Process. 85 (2) (2005) 395–399.
[14] Y. Ishikawa, R. Subramanya, C. Faloutsos, Mindreader: querying databases through multiple examples, in: Proceedings of the 24th International Conference Very Large Databases (VLDB), New York, USA, 1999, pp. 264–269.
[15] E. de Ves, J. Domingo, G. Ayala, P. Zuccarello, A novel Bayesian framework for relevance feedback in image content-based retrieval systems, Pattern Recognition 39 (2006) 1622–1632.
[16] P. Soille, Morphological Image Analysis: Principles and Applications, Springer, Berlin, 2003.
[17] A. Agresti, Cualitative Data Analysis, second ed., Wiley, New York, 2002.
[18] R Development Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2006, ISBN 3-900051-07-0.
[19] W.N. Venables, B.D. Ripley, Modern Applied Statistics with S, fourth ed., Springer, New York, 2002 ISBN 0-387-95457-0.
[20] Z. Stejic, Y. Takama, K. Hirota, Mathematical aggregation operators in image retrieval: effect on retrieval performance and role in relevance feedback, Signal Process. 85 (2005) 1297–1324.
[21] In: R. Slowinski (Ed.), Fuzzy Sets in Decision Analysis, Operations Research and Statistics, Kluwer Academic Publishers, Dordrecth, 1998.
[22] D. Filev, R. Yager, On the issue of obtaining OWA operator weights, Fuzzy Sets and Systems 94 (1998) 157–169.
[23] M. O'Hagan, Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic, in: Proceedings of 22nd Annual IEEE Asilomar Conference on Signals, Pacific Grove, CA, 1988, pp. 681–689.
[24] R. Fuller, P. Majlender, On obtaining minimal variability OWA operator weights, Fuzzy Sets and Systems 136 (2003) 203–215.

About the Author—TERESA LEÓN-MENDOZA was born in Madrid (Spain) in 1965. She graduated in Mathematics (1987) at the University of Valencia and, at the same university, received her Ph.D in Operations Research (1991). She is a Titular Professor at the Department of Statistics and Operations Research in the University of Valencia, Spain. Her present research interests are mainly concerned on fuzzy set theory, medical image analysis and functional data analysis.

About the Author—PEDRO ZUCCARELLO was born in Buenos Aires (Argentina). He graduated in Electronic Engineering in 2005 at the University of Buenos Aires. He is currently a PhD student at the University of Valencia. His research interests include audio, image and telecomunication signal processing.

About the Author—GUILLERMO AYALA was born in Cartagena (Spain) in 1962. He graduated in Mathematics (1985) and received his Ph.D in Statistics (1988), both at the University of Valencia. He is currently Titular Professor at the Department of Statistics and Operations Research in the University of Valencia, Spain. His research interests are in the areas of medical image analysis and applications of stochastic geometry in computer vision.

About the Author—ESTHER DE VES was born in Almansa (Spain) in 1971. She received the M.S. degree in Physics and Ph.D in computer Science from the University of Valencia in 1993 and 1999, respectively. Since 1994 she has been with the Department of Computer Science from the University of Valencia where is a Titular Profesor. Her current interests are in the areas of medical image analysis and image database retrieval.

About the Author—JUAN DOMINGO was born in Valencia (Spain) in 1965. He graduated in Physics in 1988 at the University of Valencia; obtained his M.Sc. in Information Technology in 1991 at the University of Edinburgh and his Ph.D. in Computing in 1993 at the University of Valencia, where he is currently Titular Professor at the Department of Informatics. His research interests include mobile robotics, medical image analysis and intelligent control.

Appendix B

International publications - Second International Conference on Computer Vision and Applications (VISAPP 2007)

Official Certificate

Pedro Zuccarello

of

University of Valencia

attended the

2nd International Conference on Computer Vision Theory and Applications (VISAPP 2007)

held in Barcelona, Spain

March 8 - 11, 2007

and presented a paper entitled:

A NOVEL RELEVANCE FEEDBACK PROCEDURE BASED ON LOGISTIC REGRESSION AND OWA OPERATOR FOR CONTENT-BASED IMAGE RETRIEVAL SYSTEM

as a conference speaker.

On behalf of the Organizing Committee,

Alpesh Ranchordas
VISAPP Conference Chair

VISAPP 2007

Proceedings of the
Second International Conference on
Computer Vision Theory and Applications

Volume IU/MTSV

Barcelona, Spain

March 8 – 11, 2007

Co-organized by
INSTICC – Institute for Systems and Technologies of Information, Control and Communication
and
CVC – Computer Vision Center

Edited by AlpeshKumar Ranchordas, Helder Araújo and Jordi Vitrià

Printed in Portugal

ISBN: 978-972-8865-74-0

Depósito Legal: 254546/07

http://www.visapp.org

secretariat@visapp.org

SHORT PAPERS

A NOVEL RELEVANCE FEEDBACK PROCEDURE BASED ON LOGISTIC REGRESSION AND OWA OPERATOR FOR CONTENT-BASED IMAGE RETRIEVAL SYSTEM

P. Zuccarello, E. de Ves
Departamento de Informtica, University de Valencia, Avda. Vicente Andrés Estellés, 1. 46100-Burjasot, Valencia, Spain
pezuc@alumni.uv.es, Esther.Deves@uv.es

T. Leon, G. Ayala
Department of Statistics and Operations Research, University of Valencia, Valencia, Spain
Teresa.Leon@uv.es, Guillermo.Ayala@uv.es

J. Domingo
Institut of Robotics, University of Valencia, Valencia, Spain
Juan.Domingo@uv.es

Keywords: Visual information retrieval,relevance feedback,logistic regresion.

Abstract: This paper presents a new algorithm for content based retrieval systems in large databases. The objective of these systems is to find the images which are as similar as possible to a user query from those contained in the global image database without using textual annotations attached to the images. The procedure proposed here to address this problem is based on logistic regression model: the algorithm considers the probability of an image to belong to the set of those desired by the user. In this work a relevance proabaility $\pi(I)$ is a quantity wich reflects the estimate of the relevance of the image I with respect to the user's preferences. The problem of the small sample size with respect to the number of features is solved by adjusting several partial linear models and combining its relevance probabilitis by means of an ordered averaged weighted operator. Experimental results are shown to evaluate the method on a large image database in term of the average number of iterations needed to find a target image.

1 INTRODUCTION

The increasing amount of information available in todays world raises the need to retrieve relevant data efficiently. Unlike text-based retrieval, where key words are successfully used to index documents, content-based image retrieval poses up-front the fundamental questions of how to extract useful image features and how to use them for intuitive retrieval (Smeulders et al., 2000). The main drawback of textual image retrieval systems, that is, the annotator dependency, would be overcome in pure CBIR systems.

Image features are a key aspect of any CBIR system. A general classification can be made: low level features (color, texture and shape) and high level features (usually obtained by combining low level features in a reasonably predefined model). High level features have a strong dependency on the application domain, therefore they are not usually suitable for general purpose systems. This is the reason why one of the most important and developed research activities in this field has been the extraction of good low level image descriptors. Obviously, there is an important gap between these features and human perception (a semantic gap). For this reason, different methods (mostly iterative procedures) have been proposed to deal with the semantic gap (Rui et al., 1998). In most cases the idea underlying these methods is to integrate the information provided by the user into the decision process. This way, the user is in charge of guiding the search by indicating his/her preferences, desires and requirements to the system. The basic idea is rather simple: the system displays a set of images (resulting from a previous search); the user selects the images that are relevant (desired images) and rejects those which are not (images to avoid) according to his/her particular criterion; the system then learns from these training examples to achieve an improved performance in the next run. The process goes on iteratively until the user is satisfied. This kind of procedures are called relevance feedback algorithms (Zhou and Huang, 2003), (de Ves et al., 2006).

A query can be seen as an expression of an information need to be satisfied. Any CBIR system aims at finding images relevant to a query and thus to the information need expressed by the query. The relationship between any image in the database and a particular query can be expressed by a relevance value. This relevance value relies on the user-perceived satisfaction of his/her information need. The relevance value can be interpreted as a mathematical probability (a relevance probability). The notion of relevance probability is not unique because different interpretations have been given by different authors. In this paper a relevance probability $\pi(I)$ is a quantity which reflects the estimation of the relevance of the image I with respect to the user's information needs. Initially, every image in the database is equally likely, but as more information on the user's preferences becomes available, the probability measure concentrates on a subset of the database. The iterative relevance feedback scheme proposed in the present paper is based on logistic regression analysis for ranking a set of images in decreasing order of their evaluated relevance probabilities.

Logistic regression is based on the construction of a linear model whose inputs, in our case, will be the image characteristics extracted from a certain image I and whose output is a function of the relevance probability of the image in the query $\pi(I)$. In logistic regression analysis, one of the key features to be established is the order of the model to be adjusted. The order of the model must be in accordance with the reasonable amount of feedback images requested from the user. For example, it is not reasonable for the user to select 40 images in each iteration; a feedback of 5/10 images would be acceptable. This requirement leads us to group the image features into n smaller subsets. The outcome of this strategy is that n smaller regression models must be adjusted: each sub-model will produce a different relevance probability $\pi_k(I)$ $(k = 1 \ldots n)$. We then face to the question of how to combine the $\pi_k(I)$ in order to rank the database according to the user's preferences. OWA (*ordered weighted averaging*) operators which were introduced by Yager in 1988 (Yager, 1988) provides a consistent and versatile way of aggregating multiple inputs into one single output.

Section 2 explains the logistic regression approach to the problem. Next, in section 3 the aggregation operators used in our work are introduced. Section 4 describes the low level features extracted from the images and used to retrieve them. An crucial part of this work, the proposed algorithm, is described in detail in section 5. After that, in section 6 we present experimental results which evaluate the performance of our technique using real-world data. Finally, in section 7 we extract conclusions and point to further work.

2 LOGISTIC REGRESSION MODEL

At each iteration, a sample is evaluated by the user selecting two sets of images: the examples or positive images and the counter-examples or negative images. Let us consider the (random) variable Y giving the user evaluation where $Y = 1$ means that the image is positively evaluated and $Y = 0$ means a negative evaluation.

Each image in the database has been previously described by using low level features in such a way that the j-th image has the k-dimensional feature vector x_j associated. Our data will consist of (x_j, y_j), with $j = 1, \ldots, k$ where x_j is the feature vector and y_j the user evaluation (1= positive and 0= negative). The image feature vector x is known for any image and we intend to predict the associated value of Y. The natural framework for this problem is the generalized linear model. In this paper, we have used a logistic regression where $P(Y = 1 \mid x)$ i.e. the probability that $Y = 1$ (the user evaluates the image positively) given the feature vector x, is related with the systematic part of the model (a linear combination of the feature vector) by means of the logit function. Generalized linear models (GLMs) extend ordinary regression models to encompass non-normal response distributions and modeling functions of the mean. Most statistical software has the facility to fit GLMs. Logistic regression is the most important model for categorical response data. Logistic regression models are also called *logit* models. They have been successfully used in many different areas including business applications and genetics. For a binary response variable Y and p explanatory variables $X_1, \ldots, X_p$, the model for $\pi(x) = P(Y = 1 \mid x)$ at values $x = (x_1, \ldots, x_p)$ of predictors is

$$logit[\pi(x)] = \alpha + \beta_1 x_1 + \ldots + \beta_p x_p \qquad (1)$$

where $logit[\pi(x)] = \ln \frac{\pi(x)}{1-\pi(x)}$. The model can also be stated directly specifying $\pi(x)$ as

$$\pi(x) = \frac{\exp(\alpha + \beta_1 x_1 + \ldots + \beta_p x_p)}{1 + \exp(\alpha + \beta_1 x_1 + \ldots + \beta_p x_p)}. \qquad (2)$$

The parameter β_i refers to the effect of x_i on the log odds that $Y = 1$, controlling the other x_j. The model parameters are obtained by maximizing the *likelihood equations*.

In the first steps of the procedure, we have a major difficulty when having to adjust a global regression model in which we take the whole set of variables into account, because the number of images (the number of positive plus negative images chosen by the user) is typically smaller than the number of characteristics. In this case, the regression model adjusted has as many parameters as the number of datum and many relevant variables could be not considered. On the other hand it is not realistic to ask the user to make a great number of positive and negative selections from the very beginning; therefore we think that the difficulty cannot be avoided in this way. In order to solve this problem, our proposal is to adjust different smaller regression models: each model considers only a subset of variables consisting of semantically related characteristics of the image. Consequently, each sub-model will associate a different relevance probability to a given image x, and we face the question of how to combine them in order to rank the database according to the user's preferences. We can see this question as an information fusion problem.

3 AGGREGATING THE RELEVANCE PROBABILITIES

Let us denote as $\pi_1(x), \pi_2(x), \ldots, \pi_n(x)$ the different relevance probabilities associated with a given image x. Each one of them has been obtained separately by using different regression models and we need to associate a final probability $\pi(x)$ by aggregating the information provided by each $\pi_j(x)$, $(j = 1 \ldots n)$. Mathematical aggregation operators transform a finite number of inputs into a single output and play an important role in image retrieval. In (Stejic et al., 2005)the authors compare the effect of 67 operators applied to the problem of computing the overall image similarity, given a collection of individual feature similarities. Their results show how important for retrieval performance the choice of the aggregation operator is. We have not used any of the 67 operators reviewed. Instead, we decided to use the so-called ordered weighted averaged (OWA) operators (Yager, 1988) since then they have been successfully applied in different areas such as decision making, expert systems, neural networks, fuzzy systems and control, etc. An OWA operator of dimension n is a mapping $f : \Re^n \to \Re$ with an associated weighting vector $W = (w_1, \ldots, w_n)$ such that $\sum_{j=1}^{n} w_j = 1$ and where $f(a_1, \ldots, a_n) = \sum_{j=1}^{n} w_j b_j$ where b_j is the j-th largest element of the collection of aggregated objects $a_1, \ldots, a_n$. The particular cases shown in table 1 can better illustrate the idea underlying OWA operators.

Notice that no weight is associated with any particular input; instead, the relative magnitude of the input decides which weight corresponds to each input. In our application, the inputs are relevance probabilities and this property is very interesting because we do not know, a priori, which set of visual descriptors will provide us with the *best* information.

Table 1: Illustrating examples of OWA aggregation values.

W	$f(a_1, \ldots, a_n)$
$(1, 0, \ldots, 0)$	$\max_i a_i$
$(0, 0, \ldots, 1)$	$\min_i a_i$
$(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})$	$\frac{1}{n} \sum_{j=i}^{n} a_i$.

As OWA operators are bounded by the max and min operators, Yager introduced a measure called *orness* to characterize the degree to which the aggregation is like an *or* (max) operation:

$$\text{orness}(W) = \frac{1}{n-1} \sum_{i=1}^{n} (n-i) w_i. \tag{3}$$

This author also introduced the concept of *dispersion* or *entropy* associated with a weighting vector:

$$Disp(W) = \sum_{i=1}^{n} w_i \ln w_i. \tag{4}$$

$Disp(W)$ tries to reflect how much of the information in the arguments is used during an aggregation based on W.

Clearly, the vector of weights W can be pre-fixed, but a number of approaches have also been suggested for determining it according to different criteria. One of the first methods developed was proposed by O'Hagan (O'Hagan, 1988). It provides us with the vector of weights for a given level of orness (optimism) which maximizes their entropy:

$$W = argmax \sum_{i=1}^{n} w_i \ln w_i$$

$$\text{subject to} \begin{cases} \alpha = \frac{1}{n-1} \sum_{i=1}^{n} (n-i) w_i, \\ \sum_{i=1}^{n} w_i = 1, w_i \in [0,1]. \end{cases}$$

This problem is not computationally easy to solve. Fuller and Majlender (Fuller and Majlender, 2003) have obtained the analytical expression of the maximum entropy weights.

Figure 1 shows the aggregation of weights for $n = 10$ obtained with the above-mentioned method for orness value $\alpha \in [0.3, 0.7]$. In this work, the aggregation weights have been computed by using this method.

4 VISUAL FEATURES

This section deals with the low level features the system uses for predicting human judgment of image

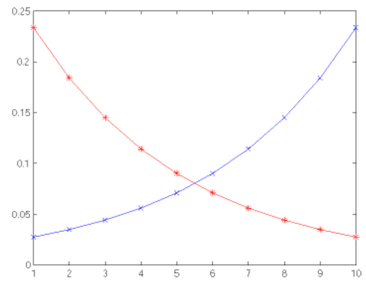


Figure 1: Aggregation weights obtained by means of Fuller and Majlender method for (orness value) $\alpha = 0.3$ (blue line) and $\alpha = 0.7$ (red line).

similarity. The relevance feedback methodology we have developed is independent of any specific image indexing and may be applied to any of them. Amongst the used characteristics we can mention:

Color representation: the current version of the system incorporates:

- A histogram of the HS (Hue, Saturation) values of the image pixels: these values are obtained after conversion to HSV color space and quantization into $(H \times S) = (10 \times 3) = 30$ color bins.

Texture representation: the system currently works with information about textures in the image:

- The granulometric cumulative distribution function. A granulometry is defined from the morphological opening of the texture using a convex and compact subset containing the origin as structuring element. In our case we have used a horizontal and a vertical segment as the structuring elements.

5 THE ALGORITHM

So far we have made a detailed description of the different theoretical components of our novel search strategy. It is now time to explain how we combine them into an efficient relevance feedback algorithm.

Let us assume a collection of images (the database) where a set of image features has been computed off-line for each image in the collection (see section 4). Let us also assume that the images are initially randomly ranked. Each iteration of the relevance feedback algorithm changes the ranking of the images according to a given set of data. By data we mean a user selection of positive and negative relevant images, and a set of aggregation weights.

An schematic description of the procedure is as follows:

Initialization: Images are randomly ranked.

Input parameters: Positive and negative relevant images are selected from among the whole collection. Let $\mathbf{I_P}$ be the set of positive samples, and $\mathbf{I_N}$ the set of negative samples. Let $W = (w_1, \ldots, w_n)$ also be the set of aggregation weights, where n is the number of relevance probabilities (outputs of the different logistic regression models) to be combined.

Logistic regression model: Using inputs selected in the previous step, several logistic regression models are fitted. Such models are applied to each image I_j in the database, obtaining their respective relevance probabilities, $(\pi_1(I_j), \ldots, \pi_n(I_j))$.

Aggregation and ranking: In order to obtain a unique relevance value, the relevance probabilities $\pi_1(I_j) \ldots \pi_n(I_j)$ should now be aggregated using the previously selected weights W (see section 3). Images are ranked according to the computed relevance values.

When a user rejects an image by selecting it as negative, we assume that the user's wish for that particular image will not change at any point in the searching process. Therefore we have implemented a memory algorithm for the selection of negative relevant images. Negative selections are remembered through all iterations. In iteration r, the set of negative relevant images used as input for the logistic regression model, $\mathbf{I_N}$, is obtained as:

$$\mathbf{I_N} = \mathbf{I_N^r} \cup \mathbf{I_N^{prev}}, \tag{5}$$

where $\mathbf{I_N^r}$ is the set of negative images selected by the user in iteration r, and $\mathbf{I_N^{prev}}$ is a subset of randomly selected images from user negative selections made in iterations 1 to $r-1$. The probability of a certain image, I_i, belonging to $\mathbf{I_N^{prev}}$ is:

$$P(I_i) = \frac{i}{\sum_{q=1}^{r-1} qN_q}, \tag{6}$$

where I_i is each image selected as negative in iteration i, N_q is the number of negative selected images in iteration q, and r is the present iteration.

6 EXPERIMENTAL RESULTS

The main objective of our algorithm is to find an image which is similar to what the user may have in mind. Therefore, the first step in the design of the experiments would be to define what is understood by "similar". Unfortunately, this is not easy since it depends on the user, and the goal of the algorithm is precisely to capture that notion of similarity that each user has, which can also change between different queries. Consequently, the valid criterion of similarity appears to be the user's opinion. This would have introduced an external variable into the experiment that would have masked the main goal: an objective evaluation of the system as such. That is why we have chosen to use an approach in which a given image has to be found. The search is considered successful if the image is ranked within the first 16. This number is arbitrary but we have checked that 16 images shown side by side is a reasonable number to localize a particular one at a first sight.

Once the criterion for termination has been adopted, the experiment will be designed by showing several images to the user; a choice of 6 images (the same for all users) was selected from a database of about 4700. These images are classified as belonging to different themes such as flowers, horses, paintings, skies, textures, ceramic tiles, buildings, clouds, trees, etc. even though the category is not used at all during the search. The 6 target images are in our experience, representative of different themes and levels of difficulty. They are displayed in figure 2.

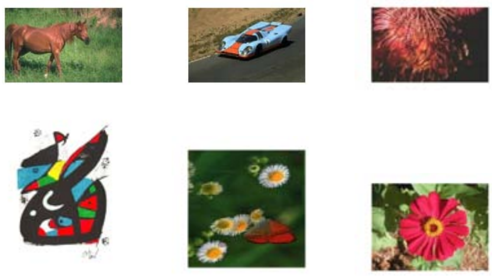


Figure 2: Target images used in experiments.

For each target image the search proceeds iteratively. In each iteration the user has to select some relevant images (similar to the target according to his/her judgment) and others significantly different from the target. The number of images of each type is left to the user, although two conditions must be fulfilled: at least one relevant and one irrelevant images must be selected and the total number of selections has to be greater than 4. The algorithm proceeds as explained in previous sections and the images are ranked. If the target appears in the first 16, it is considered to have been found; otherwise the user can move backwards or forwards to see more images in rank order and a new iteration of choosing/search/showing begins.

To ensure that the experiments are not biased, the query tasks were performed by a group of 40 users who had not been involved in the design and development of the system and had no knowledge of the content of the database or of the retrieval features and methods used (untrained users).

Table 2: Average, maximun, minimun iteration number to find a target image.

Image	It. Av.	max	min
Car	5.17(2.95)	12	1
Flower	4.17 (3.20)	17	1
Butterfly	4.71 (3.70)	19	1
firework	2.14 (1.81)	9	1
Miro	3.67 (1.55)	8	2
Glass	3.42 (1.52)	6	1
All	3.88(1.07)	19	1

Table 2 shows the average and standard deviation of the number of iterations needed to find images by these untrained users. The last row shows the average for all images and users. The experiments exhibit good performance in finding a target image (3.88 iterations in average) in the used database.

7 CONCLUSION

This paper addresses the problem of image retrieval by means of an algorithm based on logistic regression. The main advantage of the method is the facility of incorporating the feedback of the user. Its main drawback is the lack of sufficient information (too small sample) to fit the model, since the number of inputs (image features) is usually high. This has been addressed by means of partial models that get the output from each subset of the inputs. The problem of combining the information of the different models, which is a data fusion problem, is solved by using an ordered weighted averaging (OWA) operator.

Concerning the experimental results, the average number of iterations shown in 2 exhibits good performance of the procedure. Some further experimentation and results analysis is currently being carried out by our research group, where users are grouped and classified with regard to there interaction of the iterative process of image selection.

REFERENCES

de Ves, E., Domingo, J., Ayala, G., and Zuccarello, P. (2006). A novel bayesian framework for relevance feedback in image content-based retrieval systems. *Pattern Recognition*, 39:1622–1632.

Fuller, R. and Majlender, P. (2003). On obtaining minimal variability owa operator weights. *Fuzzy Sets and Systems*, 136:203–215.

O'Hagan, M. (1988). Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic. In *Proc. of 22nd Annu. IEEE Asilomar Conf. on Signals*, pages 681–689, Pacific Grove, CA.

Rui, Y., Huang, S., Ortega, M., and Mehrotra, S. (1998). Relevance feeback: a power tool for interactive content-based image retrieval. *IEEE Transaction on circuits and video technology*, 8(5).

Smeulders, A., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE transactions on Pattern Analysis and Machine Intellingence*, 22(12):1349–1379.

Stejic, Z., Takama, Y., and Hirota, K. (2005). Mathematical aggregation operators in image retrieval: effect on retrieval performance and role in relevance feedback. *Signal processing*, 85:1297–324.

Yager, R. (1988). On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Trans. Systems Man Cybernet*, 18:183–190.

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