Multimedia Systems 8: 536–544 (2003)
Digital Object Identifier (DOI) 10.1007/s00530-002-0070-3

Multimedia Systems

Relevance feedback in image retrieval: A comprehensive review

Xiang Sean Zhou[1,⋆], **Thomas S. Huang**[2]

[1] Siemens Corporate Research 755 College Road East, Princeton, NJ 08540, USA; e-mail: xzhou@scr.siemens.com
[2] Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana Champaign, 405 N Mathews Ave, Urbana, IL 61801, USA; e-mail: huang@ifp.uiuc.edu

Abstract. We analyze the nature of the relevance feedback problem in a continuous representation space in the context of content-based image retrieval. Emphasis is put on exploring the uniqueness of the problem and comparing the assumptions, implementations, and merits of various solutions in the literature. An attempt is made to compile a list of critical issues to consider when designing a relevance feedback algorithm. With a comprehensive review as the main portion, this paper also offers some novel solutions and perspectives throughout the discussion.

Key words: Relevance feedback – Content-based image retrieval – Computer vision – Classification – Pattern recognition – Small sample learning

1. Introduction

Initially developed in document retrieval during the 1960s [37, 40], relevance feedback was transformed and introduced into content-based multimedia retrieval, mainly content-based image retrieval (CBIR), during the early and mid-1990s [20,29, 34,39]. Since then, this topic has attracted tremendous attention in the CBIR community – a variety of solutions has been proposed within a short period, and it remains an active research topic today. The reasons are that *more ambiguities arise when interpreting images than words*, which makes user interaction more of a necessity; and in addition, *judging a document takes time, while an image reveals its content almost instantly to a human observer*, which makes the feedback process faster and more sensible for the end user.

A fundamental difference of relevance feedback in image retrieval as opposed to document retrieval is that the latter is based on symbolic representations, with direct mapping to human interpretations; while for images, a precise high-level symbolic representation is hard to extract automatically; and the extractable low-level features (e.g., color, texture, shape, etc.) are often inadequate or even misleading for high-level perception-based retrieval tasks. In a nutshell, "an image is worth a thousand words", and the machine does not know what these words are.

Even if we assume that the low-level features are somewhat correlated with the high-level semantics, we still need a user-in-the-loop, because images reside in a continuous representation space, in which semantic concepts are best described in *discriminative subspaces* – "cars" are of certain *shape* while "sunset" is more describable by *color*. In other words, only a small subset of features (or a subspace of the original space) is active for describing any giving concept. More importantly, different users at different times may have different interpretations or intended usages for the same image, which makes off-line, user-independent learning undesirable in general. Fully automated off-line preprocessing (e.g., clustering, classification) makes sense for some specific applications with well-defined image classes. However, for many others, *the* best answer does not exist, and ignoring a user's individuality can be as senseless as trying to determine the world's greatest color.

A straightforward way of getting the user into the loop is to ask the user to tune system parameters during the retrieval process, but it is too much of a burden for a common user. A more feasible form of interaction is to ask the user to provide feedback regarding the (ir)relevance of the current retrieval results. The system then learns from these training examples to achieve an improved performance in the next round, iteratively if necessary.

Relevance feedback algorithms have been shown to provide dramatic performance boost in retrieval systems [18,27, 29,34,39,46,51,52,55,62].

2. The relevance feedback problem

Since the general assumption is that every user's need is different [20] and time varying, a database cannot adopt a fixed clustering structure; and *the total number of classes* and *the class membership* are not available beforehand, since these are assumed to be user-dependent, and time varying as well. Of course, these rather extreme assumptions can be relaxed in a real-world application to the degree of choice. (For more arguments, see Sect. 4.3.)

A typical scenario for relevance feedback in content-based image retrieval is as follows:

⋆ Work was done while at the University of Illinois.

Step 1. Machine provides an initial retrieval results, through query-by-keyword, sketch, or example, etc.

Step 2. User provides a judgment on the currently displayed images as to whether, and to what degree, they are relevant or irrelevant to her/his request.

Step 3. Machine learns and tries again. Go to step 2.

If each image/region is represented by a point in a feature space, relevance feedback with only positive (i.e., relevant) examples can be cast as a density estimation [19,28] or novelty detection [7,43] problem. While with both positive and negative training examples it becomes a classification problem, or an online learning problem in a batch mode, but with the following characteristics:

Small sample issue. The number of training examples is small (typically < 20 per round of interaction, depending upon the user's patience and willingness to cooperate) relative to the dimension of the feature space (from dozens to hundreds, or even more), while the number of classes is large for most real-world image databases. For such small sample sizes, some existing learning machines such as support vector machines (SVM) [50] cannot give stable or meaningful results [46,62], unless more training samples can be elicited from the user [47].

Asymmetry in training sample. The desired output of information retrieval is not necessarily a binary decision on each point as given by a classifier, but rather a rank-ordered top-k returns. This is a less demanding task, since the rank or configuration of irrelevant classes/points is of no concern as long as they are well beyond the top-k returns. Most classification or learning algorithms (e.g., discriminant analysis [10] or SVM [50]) treat positive and negative examples interchangeably, and assume that both sets represent the true distributions equally well. However, in reality, the small number of negative examples is unlikely to be representative for all the irrelevant classes; thus, an asymmetric treatment may be necessary [62].

Real time requirement. Finally, since the user is interacting with the machine in real time, the algorithm should be sufficiently fast, and if possible avoid heavy computations over the whole dataset.

It should be noted that the above discussion covers some but not all the scenarios and proposals in the literature. For example, user feedback may take the form of a "comparative judgment" [8] instead of a class label; and local image matching or object detection may be better accomplished by using multiple high-dimensional histograms or mixture models as image descriptors, instead of using just one feature vector (see the next section for a comprehensive review on existing algorithms).

3. Variants of relevance feedback algorithms

It is not the intention of this section to list *all* the existing techniques, but rather to point out major variants and compare their merits. We would emphasize that *under the same notion of "relevance feedback", different methods might have adopted different assumptions or problem settings, and are thus incomparable*. The following lists some of the conceptual dimensions along which some schemes differ greatly from others. These can be separated into two broad classes; one includes several aspects of the user behavior model (Sects. 3.1–3.3), and the other covers algorithmic assumptions and alternatives (Sects. 3.4–3.7).

3.1. User model: What to look for?

While most of the work assumes the user is looking for "a class of similar items" to the query at hand (*"category search"*), Cox et al. [8,9] assume that the user is looking for "a particular target item" (*"target search"*) and that the feedback is in the form of "relative judgment", i.e., positive images are not necessarily the target, but "closer" to the target than others. A Bayesian framework is adapted to estimate an updated probabilistic distribution over all the test images in the database after each round of user interaction, until the target appears in the set of displayed images. The user model is assumed (arguably) to be sigmoidal in distance, reflecting the heuristic that images closer to the selected (positive) images than "nonselected" ones are more likely to be the target.

Note that in reality, *user consistency* is hard to achieve, i.e., it is often difficult for a user to tell between two images which is "closer" to a third one *consistently* in accordance with the underlying feature representations adopted by the machine. In the light of this difficulty, the user modeling has to be "soft", or *probabilistic* in nature [9].

While searching a large image database for a specific target, it is expected that in general more than one round of user interaction is needed. The machine then faces the question of "how to select the best set of images for each round to ask for user feedback so that the total number of iterations needed to reach the target is minimal?" [9]. This issue is further elaborated in Sect. 3.3.

3.2. User model: What to feed back?

Some algorithms assume the user will give a binary feedback for positive and negative examples [32,46,47]; some only take positive examples [7,38]; some take positive and negative examples with "degree of (ir)relevance" for each [39,62]; some assumes the feedback is only a "comparative judgment" instead of a definite hit or miss [9]; some uses both labeled and unlabeled data for training: Wu et al. [58] proposed the D-EM algorithm within a *transductive learning* framework, and used examples from user feedback (labeled data) as well as other data points (unlabeled data). It performs discriminant analysis inside EM iterations to select a subspace of features, such that the two-class (positive and negative) assumption on the data distributions has better support. The results were promising, but computation can be a concern for large datasets.

A novel form of training is "learning from layout of images" during browsing or the data visualization process [30, 41]. The idea is to ask the user to layout images on a "table" (i.e., a 2D space, which can be obtained using multidimensional scaling, or MDS techniques), or to manipulate an existing 2D layout of images, according to the user's interpretation

of the semantic relationships among images. The machine is expected to layout other images in a similar fashion after learning. The learning can proceed by finding a feature-weighting scheme under which a principle component analysis (PCA) will yield a layout of the training images that is most similar to the user's layout. The weights are then applied to test images, and the PCA is used to splat ("spread flat") the test images for a 2D image layout [30].

3.3. User model: Greedy vs. cooperative

If we assume that the user is greedy and Impatient, and thus expects the best possible retrieval results after each feedback round, the machine should always display the *most-positive images* based on previous training. In this case, the user can terminate the query process at any point in time, and will always get the best results so far. Additional user feedback or "training", if any, will be performed on these *most-positive images*. This is the strategy adopted by most, if not all, early relevance feedback schemes.

However, for applications where the user is co-operative and willing to provide more than one screen of training samples before seeing the results, a new question arises: "After getting feedback for one or more screens of training images, which of the remaining images shall the machine select to ask the user to label in order to achieve the highest information gain?"

The key to understanding this problem is to realize that, from the machine's point of view, "selecting 40 examples in one batch for user labeling and training" is not as good as "selecting 20 first, training on them and then selecting another 20 based on what has been learned."

In general, the *most-informative images* [8] will not coincide with the *most-positive images*, since some of the latter might already be labeled, or tend to be very correlated with images with known labels, thus providing less new information. Intuitively, the *most-informative images* should be those whose labels the learner is most uncertain about.

As shown in Fig. 1, we have also dubbed these two scenarios the *show-me-the-results* and *ask-me-questions* user models, respectively.

Active learning [2], or selective sampling [13], studies the strategy for the learner (i.e., the machine) to actively select samples to *query*[1] the teacher (i.e., the user) for labels, to achieve the maximal information gain, or the minimized entropy/uncertainty in decision-making. Its early application for document classification can be found in Lewis and Gale [23]. Recent applications in image retrieval can be found in Cox et al. [8], Li et al. [24], and Tong and Chang [47].

Cox et al. [8] used Monte Carlo sampling in search of the set of images that, once labeled, will minimize the *expected* number of future iterations. In estimating the expected number of future iterations, entropy is used as an estimate of the number of questions to be asked under the ambiguity specified by the current probability distribution of the target image over all the test images.

[1] A term used in the active learning literature to denote the action by the learner (i.e., the machine) to ask for training data from the teacher (i.e., the user). This should not be confused with the *query* concept used in information retrieval.

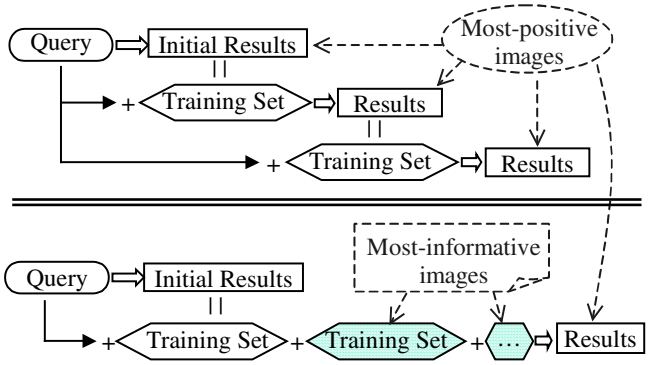

Fig. 1. *Top*: the *show-me-the-results* scenario for a greedy user, where further training, if any, will be performed on the current best results. *Bottom*: the *ask-me-questions* scenario for a co-operative user, where the machine can actively select more than one screen of samples to be added *sequentially* into its training set

Tong and Koller [48] and Tong and Chang [47] proposed the SVM active learning algorithm for applications in text classification and image retrieval [50]. The aim is to select the item(s) to maximally reduce the size of the version space in which the class boundary lies. Without knowing a priori the label of the candidate, the best strategy is to halve the version space each time. They attempted to justify that selecting the points near the SVM boundary can approximately achieve this goal, and it is more efficient than other, more sophisticated schemes, which require exhaustive trials on all the test items. Therefore, in their work, the points near the SVM boundary are used to approximate the *most-informative points*; and the *most-positive images* are chosen as those furthest from the boundary on the positive side in the *feature space*[2] [50].

Note that the differences between Cox et al. [8] and Tong and Chang [47] are not only in the analyzing tools they use, but also in the problem settings they assume: the former looks for a target image, while the latter searches for a classifier. (Though the two scenarios may overlap at extreme cases.)

Finally, there is no reason why we cannot mix the *most-informative* and *most-positive* images on one screen [46] – the question is how do we strike a balance between the two optimally (in a sense, e.g., by maximizing a confidence measure of retrieval performance)?

3.4. Algorithmic assumptions: Feature selection and representation

In terms of feature selection, while most CBIR systems use traditional image features such as color histogram or moments, texture, shape, and structure features, there are alternatives. Tieu and Viola [46] used more than 45,000 "highly selective features", and a boosting technique to learn a classification function in this feature space. The features were demonstrated to be sparse with high kurtosis, and were argued to be expressive for high-level semantic concepts. Weak two-class classifiers were formulated based on a Gaussian assumption for

[2] A term used in the kernel machine literature to denote the new space after the nonlinear transform implied by the kernel – this should not be confused with the *feature space* concept otherwise used to denote the image representation space.

both positive and negative (randomly chosen) examples along each feature component, independently. The strong classifier is a weighted sum of the weak classifiers as in AdaBoost [12].

As for feature representation, while most assume one feature vector per image/region as the basic representation, Vasconcelos and Lippman [51] adopted a Gaussian mixture model on DCT coefficients as the image representation. Bayesian inference is then applied for classification and learning over time. Richer information captured by the mixture model also makes image regional matching possible.

3.5. Algorithmic assumptions: Class distribution

Another issue is what distribution to impose on the target class(es). Gaussian assumption is a common and convenient choice [19,38]. A specific form of nonlinear distribution is the so-called "disjunctive set" or "multimode" distribution, which has been addressed by a number of researchers in various ad hoc ways [5]. Wu et al. [58] treated multiple queries as a disjunctive set, and used an aggregate dissimilarity function to combine for a candidate image the pair-wise distances to every positive example as the distance measure. This should be compared to a Parzen window method [28], in which Parzen window density estimation was applied to capture nonlinearity in the distribution of positive examples. A principled way to deal with nonlinearity is to use reproducing kernel-based algorithms. A kernel-based one-class SVM as the density estimator for positive examples was shown in Chen et al. [7] to outperform whitening transform-based linear/quadratic methods. BiasMap [62] and the SVM active learning algorithm [47] both adopt the kernel form to cope with nonlinear distributions, with the former emphasizing the small sample issue, while the latter explores the active learning aspect. It is worth noting that most of the above algorithms use the RBF kernel or Gaussian kernel, which has the "over-fitting" problem. The selection of kernel parameters can be tricky, and is under active investigation.

3.6. Data structure

If a hierarchical tree structure is adopted in a database for more efficient access [6], learning becomes more difficult, since the tree structure needs to be updated after new knowledge is discovered through the user interaction. To efficiently update such a tree structure, the trade-off offered by Chen et al. [6] between speed and accuracy for searching becomes crucial. However, in any case, the reorganization of a hierarchical structure (such as a similarity pyramid) for a large image database is still a stunning task to perform, and perhaps should only be carried out once in a while. The question is: how to update the tree structure according to the user's understanding of similarity? This is similar to the problem of "learning the relative feature importance from a layout of images", as mentioned in Sect. 3.2. One approach is to find the set of feature weights under which the clustering behaviors among training images can best approximate those provided by the user. Using the newly weighted features as the representation, all test images can be re-clustered, hopefully in a way reflecting the user's understanding and preference.

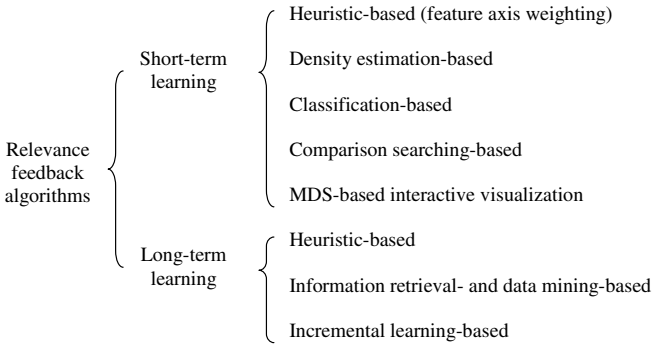

Fig. 2. A nonexhaustive taxonomy of relevance feedback algorithms

3.7. Objective functions and learning machines

In this direction lies the greatest variability among different methods. A number of early methods – self-labeled as "relevance feedback" [6,32,33,39,41] or not [25] – propose to learn a new query and the relative importance of different features or feature components, while others learn a linear transformation in the feature space, taking into account correlations among feature components [19,38,62]. Some of the latest work treats it either as a density estimation [7,58], learning [27,46,48,47], or classification [51,58] problem. A nonexhaustive taxonomy is show in Fig. 2. In this section, we discuss mainly short-term learning. Long-term learning is discussed in Sect. 4.4. We should point out that some of the categories are not yet fully explored in the literature, so this should be interpreted as the authors' proposed future research directions; for example, the use of incremental learning principles in relevance feedback.

In its short history, relevance feedback developed along the path from heuristic-based techniques to optimal learning algorithms, with early work inspired by term weighting and relevance feedback techniques in document retrieval [40]. These methods proposed heuristic formulation with empirical parameter adjustment, mainly along the line of independent axis weighting in the feature space [33–35,39,41]. The intuition is to emphasize more those feature(s) that best cluster positive examples and separate the positive and negative.

Early work [34,39] had clear roots in the document retrieval field. For example, in Rui et al. [39], learning based on "term frequency" and "inverse document frequency" in the text domain was transformed into learning based on ranks of the positive and negative images along each feature axis in the continuous feature space. Picard et al. [34] quantized the features and then grouped images or regions into hierarchical trees whose nodes were constructed through single-link clustering. Groupings were then weighted using set operations.

Kohonen's *learning vector quantization* (LVQ) algorithm [54] and tree-structured *self-organizing map* (TS-SOM) [21] were used for dynamic data clustering during relevance feedback. Laaksonen et al. [21] used TS-SOMs to index images along different feature axes such as color and texture. Positive and negative examples were mapped to positive and negative impulses on the maps, and a low-pass operation on the maps was argued to *implicitly* reveal the relative importance of different features, because a "good" map will keep positive examples cluster while negative examples scatter away. This was

based on a similar intuition as that of Peng et al. [33], where a probabilistic method was used instead to capture feature relevance. The assumption of feature independence imposed in these methods is rather artificial.

Later on, researchers began to look at this problem from a more systematic point of view by formulating it into an optimization, learning, or classification problem. In Ishikawa et al. [19] and Rui and Huang [38], based on the minimization of total distances of positive examples from the new query, the optimal solutions turned out to be the weighted average as the new query and a whitening transform (or Mahalanobis distance metric) in the feature space. Additionally, Rui and Huang [38] adopted a two-level weighting scheme to better cope with singularity issue due to the small number of training samples. To take into account negative examples, Schettini et al. [42] updated feature weights along each feature axis by comparing the variance of positive examples to the variance of the union of positive and negative examples.

MacArthur et al. [27] cast relevance feedback as a two-class learning problem, and used a decision tree algorithm to sequentially "cut" the feature space until all points within a partition are of the same class. The database was classified by the resulting decision tree: images that fall into a relevant leaf were collected and nearest neighbors of the query were returned.

Some of the approaches are intended for offline learning, but have the potential for online implementation. For example, Guo et al. [15] used AdaBoost for face recognition and audio retrieval. In 2001, a constrained majority voting (CMV) strategy was proposed to speed up pair-wise comparisons for multi-class classification. Note that, in their case, labeled training samples are available for all classes.

There were also schemes for learning object structure from examples based on image segmentation. Xu et al. [59] proposed a hierarchical formation scheme for object description from elementary regions determined by a segmentation using color and edge. From examples, the system learns a "composite node" of several regions with an adjacency matrix representing their spatial relationships. Ratan et al. [36] used a *multiple-instant learning* model to learn the most important subimage(s) from example images, which are represented as a bag (collection) of instances (subimages). The adopted *Diverse Density algorithm* tries to find the area in feature space that is shared by all positive images while far from all negative subimages. Along the same line is the work by Forsyth and Fleck [11], where the system learns a "body plan" for object. Hong and Huang [17] defined an object (or scene) as a contextual pattern and adopted an ARG (attributed relational graph) [49] to represent it. They developed an automatic contextual pattern modeling scheme, which learns a probabilistic pattern ARG model from multiple sample ARGs via the EM algorithm. The learned pattern ARG model captures the probabilistic characteristics of both the appearance and the structure of the object, which may be observed under changing conditions, and may only occupy portions of the training images and can be partially occluded. The concern is on the computational complexity, which is still far beyond the real-time requirement of relevance feedback.

4. Issues to consider when designing a relevance feedback algorithm

In the following, we try to compile a list of critical issues to consider when designing or selecting a relevance feedback algorithm. These are intended to be common issues across various applications or user assumptions.

4.1. Negative examples

How to treat the *small number* of negative examples may be the central issue when negative feedback is to be considered. Tieu and Viola [46] used random sampling to get around the small sample issue, taking a risk of treating positive points as negative training samples. Vasconcelos and Lippman [52] assumed that a negative example for class S_i shall be a positive example for the complement of class S_i, and quantified in terms of likelihoods as follows:

$$P(\bar{y}|S_i = 1) = P(y|S_i = 0) \tag{1}$$

where $\bar{y}$ means that y is treated as a negative example. Special steps are needed to avoid over-emphasizing the importance of negative examples.

Nastar et al. [32] proposed empirical formulae to take into account negative examples while estimating the distribution of positive examples along each feature component. Another ad hoc formulation was proposed by Brunelli and Mich [5]. Zhou and Huang [60,62] used the intuition that "all positive examples are alike in a way, each negative example is negative in its own way", and proposed an asymmetric treatment for positive and negative examples: they assumed that positive examples have a compact low-dimensional support, while negative examples can have any configuration. A custom designed discriminant analysis, namely, biased discriminant analysis (BDA), is applied to find the transformed, reduced-dimension space where positive examples cluster while the negative scatter away. This scheme can be regarded as a "discriminative whitening transform". The proposed kernel form, namely, BiasMap, can handle nonlinear configurations (e.g., multimode for the positive distribution) in a principled way.

As a side note, it would be interesting to explore the possibility of incorporating negative examples in learning object structure from examples [11,17,36,59].

4.2. Singularity issue in sample covariance matrix

Many relevance feedback algorithms make use of the sample covariance matrix and its inverse [19,24,39,42,62]. When the number of training examples is smaller than the dimensionality of the feature space, the singularity issue arises. The substitution of the *Moore–Penrose inverse* or *pseudo-inverse matrix* for the regular inverse proposed in Ishikawa et al. [19] is not only mathematically unfounded, but also counter-intuitive: Imagine a diagonal covariance matrix with the ith diagonal element being 0; according to the "weight by the inverse of the variance" heuristic implied in this method, this indicates that the ith axis of the feature space is very expressive, thus it will receive a very high weight. However, Ishikawa [19] will put a weight of zero on the i axis.

Another proposal was to adopt a hierarchical weighting scheme (assuming a *block diagonal* matrix instead of the full covariance matrix) so that fewer parameters need to be estimated, and in the extreme case, just use a diagonal matrix [38]. This implies a forced independence assumption. Although intuitively appealing – the independence assumption between color and texture in some cases may be valid – for cases in which this assumption does not hold, the block-diagonal or diagonal treatment can yield extremely biased eigenvalue estimations. As an example, assuming two positive examples $[0,0]^T$ and $[1,1]^T$, the sample covariance matrix C is

$$C = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}; \quad C_{\text{diag}} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}. \tag{2}$$

Only estimating the diagonal matrix C_{diag} will result in equal weighting of the two axes by the weight vector $[2,2]^T$ (ignore normalization). While an apparently better solution is to rotate the space clockwise by $45°$, and then weight the vertical axis more. Additionally, invertibility is still not guaranteed, only diagonal elements are estimated.

It is known that under a small sample, the sample covariance matrix is statistically biased, in the sense that the large eigenvalues become larger and the small become smaller. A statistically valid solution is to add regularization terms on the diagonal of the sample covariance matrix before the inversion, which is also known as a "linear shrinkage estimator" [16,14, 62]:

$$\hat{C} = (1-\mu)C + \frac{\mu}{n} tr[C] I. \tag{3}$$

Here $tr[C]$ denotes the trace of C, n is the dimension of the feature space, and $0 < \mu < 1$ controls the amount of shrinkage toward the identity matrix I. μ can be set as a function of the number of training examples: the smaller the training sample, the larger is μ.

This operation, although simple and seemingly unintentional, actually compensates the aforementioned bias [16,14]. Following the example above:

$$\begin{aligned} \hat{C} &= \begin{bmatrix} 0.5+0.01 & 0.5 \\ 0.5 & 0.5+0.01 \end{bmatrix}; \quad \hat{C}^{-1} = V \Lambda V^{-1}; \\ V &= \begin{bmatrix} -0.707 & -0.707 \\ -0.707 & 0.707 \end{bmatrix}; \quad \Lambda = \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}. \end{aligned} \tag{4}$$

The solution in Eq. (4) provides the rotation of $45°$ (by V), followed by a proper weighting of the axes (by Λ).

At this point it is worth noting that a unifying view of relevance feedback algorithms can be of "learning an optimal transform in the feature space", because: when only positive examples are considered, the "generalized ellipsoid distance metric" [19] is equivalent to a whitening transform followed by the Euclidean metric [38], since the eigenvalues are also the singular values; when negative examples are considered using discriminant analysis, as in Zhou and Huang [62], the generalized eigenvalues are not the same as the singular values, and the solution is a generalized whitening transform or discriminative whitening transform followed by the Euclidean metric.

Feature normalization

Different image features need to be normalized to have comparable statistics, say normal distribution. (A set of alternatives is discussed in Aksoy and Haralick [1].) Not surprisingly, this normalization can also be extended to a transformation of the feature space. From a discrimination point of view, the optimal normalization shall be the transform that separates *all* the semantically meaningful classes in the dataset from clusters within each class. Since the class membership is not known a priori, one possible solution is to use the accumulated feedback from all the users as the training set to be fed into a multiple discriminant analysis [9,10] framework to yield a transform that is optimal for the training data available so far. This implies more computation, but can give better performance than the straightforward normal distribution assumption. A simplified example is that if all users emphasize *color feature* more than *texture* in *all* cases, then there is no reason for maintaining equal variances along these two axes – stretching the *color* axis can give a better initial retrieval result. This is illustrated in Fig. 3. Note that the "stretching direction" does not have to be aligned with the original axes as shown in the figure – correlations among axes can be modeled as well.

4.3. Pre-clustering and long-term learning

It may be argued that unsupervised clustering techniques – EM using minimum description length criteria, or mean shift – can determine the number of clusters in the database offline, automatically. However, semantically meaningful clustering depends upon the subspace in which a semantic concept class lies: an image of a "white horse" in the feature space is not necessarily closer to a "red horse" than it is to a "white sheep", unless a proper discriminating subspace (say, discounting *color*) can be specified beforehand – which is, however, exactly what relevance feedback is trying to learn in the first place. So *in principle, the rationale of relevance feedback contradicts that of pre-clustering*. It is even more so when differences in perception and interpretation among different users at different times are taken into account – the clustering structure of a database changes for different users at different times.

However, in practice, prior domain knowledge – if it holds true for all users of the system – can be used to guide a preorganization of the dataset. This can be the case for some applications such as medical image databases, for which semantically meaningful *static* clusters exist and can be identified offline to improve the real time performance. In such cases, knowledge can also be "accumulated" during user interaction from time to time, and from user to user, and we refer to this as "long-term learning" (see Fig. 2) [4,22,29,45].

In Bartolini et al. [4], the authors have assumed the existence a *static* mapping from each point to an "optimal" query point (the cluster center) and an "optimal" distance function (the shape of the cluster). This mapping is learned across time and across different users, and is then used to "bypass" subsequent relevance feedback loops.

Su et al. [45] used incremental updating formulae across relevance feedback sessions to efficiently estimate class-specific Gaussian parameters in the PCA subspace.

Overall, in considering the pre-clustering or long-term, across-user learning issue, a trade-off has to be made between

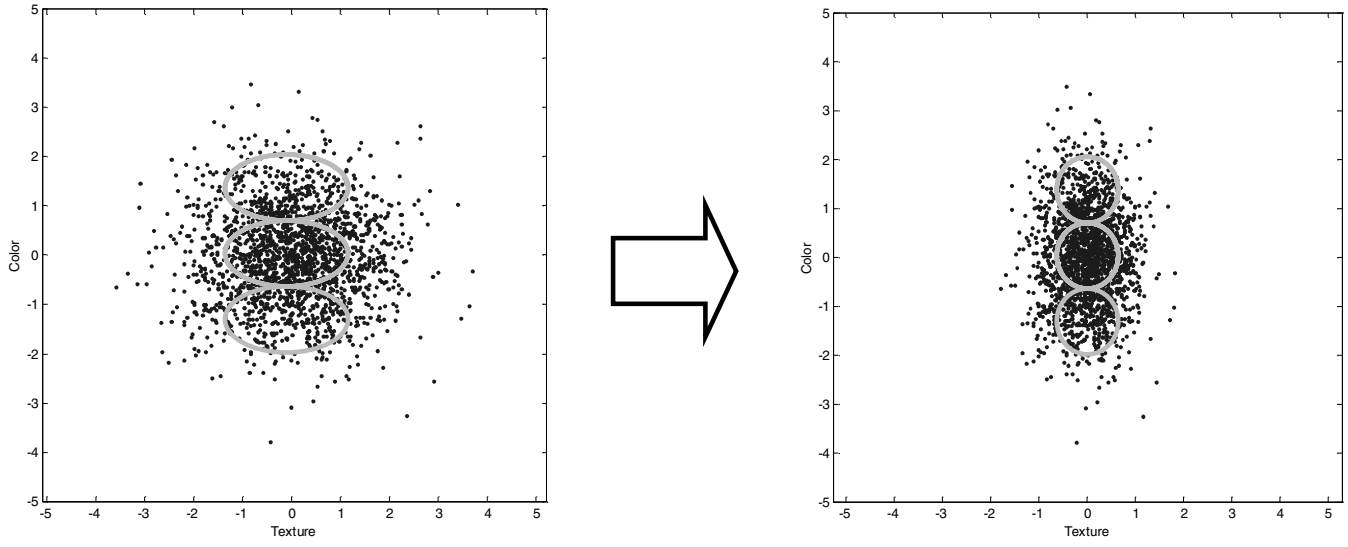

Fig. 3. Feature normalization as a discriminating transformation: assuming all meaningful classes have ellipsoidal contours as shown

flexibility (in supporting *individuality* online) and efficiency (by storing common knowledge offline).

4.4. Global vs. regional query

Most relevance feedback schemes are designed to deal with global image features only, which apparently is not the best choice. Some algorithms can be extended to deal with image blocks using a *concatenated* feature vector as the representation, and a hierarchical weighting scheme automatically reveals the relative importance of different image blocks during user interactions [61]. Vasconcelos and Lippman [52] used Bayesian inference on image local features for relevance feedback learning. This scheme is inherently capable of image regional query, given properly learned priors. Explicit local object learning and modeling schemes [11,17,29,31,36,59, 64], if robust and fast enough, could be the ultimate choice to achieve the highest performance for image regional query through relevance feedback.

4.5. Incorporating textual annotations

The so-called "semantic gap" between high-level concepts in the user's mind and low-level features extracted by the machine is so wide in many cases that the use of keywords or annotations (where available) is key to meaningful retrieval. The research efforts in combining low-level features with text include joint use of textual and visual features for querying and relevance feedback learning; learning of visual models for concept classes annotated by keywords; and the learning of keyword relations from relevance feedback [3,26,44,63].

4.6. Complexity of a nearest neighbor search

When the size of the dataset is large and the dimensionality of the representation space is high, even a simple nearest neighbor search (under changing distance metric) can be computationally formidable for real time performance. One solution can be an adaptive nearest neighbor search [57], which updates a relatively small number of nearest neighbors intelligently and efficiently from one iteration to the next without searching the whole dataset repeatedly. Other solutions exploit hierarchical data structures as well as parallel processing architectures to speed upp the nearest neighbor search [53].

A challenging problem, as mentioned before, is how to dynamically update a hierarchical data structure according to user fed-back information.

5. Summary

Targeted at a very specific application scenario, namely *the real-time learning from user interactions during information retrieval*, relevance feedback as a classification or learning problem possesses very unique characteristics and difficulties. A successful algorithm is one that is tailored to address these special issues.

In this paper, we have compared and analyzed a variety of relevance feedback algorithms in the literature, most of which are from the content-based multimedia retrieval research area, with some from other areas, but having the essence of – or the potential of being used as – a relevance feedback algorithm during information retrieval.

One of the key observations is that, even though labeled the same as "relevance feedback" algorithms, many schemes were developed under quite different application or user assumptions. We highlight these differences, and compare their merits. Through the comparison and analysis of existing literature, we have discovered some common problems across different approaches, as well as some misconceptions; a list of such critical issues is presented and elaborated upon in the hope of aiding readers' efforts in designing fast and effective relevance feedback algorithms.

Some future research directions were proposed throughout the discussion.

Acknowledgements. This work was supported in part by NSF Grant CDA 96-24396. Comments and suggestions from the reviewers were greatly appreciated, and have enriched the content of this paper.

References

1. Aksoy S, Haralick RM (2000) Probabilistic vs. geometric similarity measure for image retrieval. In: IEEE Conf. Computer Vision and Pattern Recognition, South Carolina
2. Angluin D (1988) Queries and concept learning. Mach Learn 2(3):319–342
3. Barnard K, Forsyth DA (2001) Learning the semantics of words and pictures. In: International Conf. on Computer Vision, Vancouver, Canada
4. Bartolini I, Ciaccia P, Waas F (2001) FeedbackBypass: A new approach to interactive similarity query processing. In: International Conf. on Very Large Data Bases (VLDB), Rome, Italy
5. Brunelli R, Mich O (2000) Image retrieval by examples. IEEE Trans Multimedia 2(3):164–171
6. Chen J-Y, Bouman CA, Dalton J (2000) Hierarchical browsing and search of large image databases. IEEE Trans Image Process 9(3):442–445
7. Chen Y, Zhou XS, Huang TS (2001) One-class SVM for learning in image retrieval. In: International Conf. on Image Processing, Greece
8. Cox IJ, Miller M, Minka TP, Papathomas T, Yianilos P (2000) The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. IEEE Trans Image Process 9(1):20–37
9. Cox IJ, Miller M, Minka TP, Yianilos P (1998) An optimized interaction strategy for Bayesian relevance feedback. In: IEEE Conf. Computer Vision and Pattern Recognition, Santa Barbara, CA
10. Duda RO, Hart PE (1973) Pattern classification and scene analysis. Wiley, New York
11. Forsyth DA, Fleck MM (1997) Finding people and animals by guided assembly. In: International Conf on Image Processing, Santa Barbara, CA
12. Freund Y, Schapire RE (1999) A short introduction to boosting. J Japan Soc Artif Intell 14(5):771–780
13. Freund Y, Seung HS, Shamir E, Tishby N (1993) Selective sampling using the query by committee algorithm. Advances in neural information processing systems. MIT Press, Cambridge, MA
14. Friedman J (1989) Regularized discriminant analysis. J Am Stat Assoc 84(405):165–175
15. Guo G, Zhang HJ, Li SZ (2001) Boosting for content-based audio classification and retrieval: an evaluation. Microsoft Research Technical Report: MSR-TR-2001-15
16. Haff LR (1980) Empirical Bayes estimation of multivariate normal covariance matrix. Ann Stat 8:586–597
17. Hong P, Huang TS (2001) Spatial pattern discovering by learning the isomorphic sub-graph from multiple attributed relation graphs. In: 8th International Workshop on Combinatorial Image Analysis, PA
18. Hong P, Tian Q, Huang TS (2000) Incorporate Support Vector Machines to Content-Based Image Retrieval with Relevance Feedback. In: IEEE 2000 International Conference on Image Processing, Vancouver, Canada
19. Ishikawa Y, Subramanya R, Faloutsos C (1998) MindReader: query databases through multiple examples. In: International Conf. on Very Large Data Bases (VLDB), NY
20. Kurita T, Kato T (1993) Learning of personal visual impression for image database systems. In: International Conf. Document Analysis and Recognition
21. Laaksonen J, Koskela M, Oja E (1999) PicSOM: Self-organizing maps for content-based image retrieval. In: INNS-IEEE International Joint Conference on Neural Networks, Washington, DC
22. Lee C, Ma WY, Zhang HJ (1998) Information embedding based on user's relevance feedback for image retrieval. In: SPIE Photonics East
23. Lewis DD, Gale WA (1994) A sequential algorithm for training text classifiers. In: ACM-SIGIR Conf. R&D in Information Retrieval, Dublin, Ireland
24. Li B, Chang E, Li C (2001) Learning image query concepts via intelligent sampling. In: International Conf. on Multimedia and Exposition, Tokyo, Japan
25. Lowe D (1995) Similarity metric learning for a variable-kernel classifier. Neural Computation 7(1):72–85
26. Lu Y, Hu C, Zhu X, Zhang HJ, Yang Q (2000) A unified framework for semantics and feature based relevance feedback in image retrieval systems. In: ACM Multimedia Conference, CA
27. MacArthur SD, Brodley CE, Shyu C (2000) Relevance feedback decision trees in content-based image retrieval. In: IEEE Workshop CBAIVL, South Carolina
28. Meilhac C, Nastar C (1999) Relevance feedback and category search in image databases. In: IEEE Int. Conf. on Multimedia Computing and Systems, Italy
29. Minka TP, Picard RW (1996) Interactive learning using a 'society of models'. In: IEEE Int. Conf. Computer Vision and Pattern Recognition
30. Moghaddam B, Tian Q, Lesh N, Shen C, Huang TS (2001) Visualization and layout for personal photo libraries. In: International Workshop on Content-based Multimedia Indexing, Italy
31. Moghaddam B, Zhou XS (2002) Factorized local appearance models. In: International Conf. on Pattern Recognition, Quebec City, Canada
32. Nastar C, Mitschke M, Meilhac C (1998) Efficient query refinement for image retrieval. In: IEEE Conf. Computer Vision and Pattern Recognition, CA
33. Peng J, Bhanu B, Qing S (1999) Probabilistic feature relevance learning for content-based image retrieval. Comput Vision Image Understanding 75:150–164
34. Picard RW, Minka TP, Szummer M (1996) Modeling user subjectivity in image libraries. In: International Conf. on Image Processing, Lausanne, Switzerland
35. Porkaew K, Mehrotra S, Ortega M (1999) Query reformulation for content based multimedia retrieval in MARS. In: IEEE Int. Conf. Multimedia Computing and Systems
36. Ratan AL, Grimson MOW, Lozano-Perez T (1999) A framework for learning query concepts in image classification. In: IEEE Conf. Computer Vision and Pattern Recognition, CO
37. Rocchio JJ (1966) Document retrieval system–optimization and evaluation. PhD dissertation, Harvard Computational Lab, Harvard University, Cambridge, MA
38. Rui Y, Huang TS (2000) Optimizing learning in image retrieval. In: IEEE Conf. Computer Vision and Pattern Recognition, South Carolina
39. Rui Y, Huang TS, Ortega M, Mehrotra S (1998) Relevance feedback: A power tool in interactive content-based image retrieval. IEEE Trans Circuits Syst Video Technol 8(5):644–655
40. Salton G (1989) Automatic text processing. Addison-Wesley, Reading, MA
41. Santini S, Jain R (2000) Integrated browsing and querying for image database. IEEE Trans Multimedia 7(3)
42. Schettini R, Ciocca G, Gagliardi I (1999) Content-based color image retrieval with relevance feedback. In: International Conf on Image Processing, Kobe, Japan

43. Scholkopf B, Williamson R, Smola A, Shawe-Taylor J, Platt J (2000) Support vector method for novelty detection. In: Advances in neural information processing systems. MIT Press, Cambridge, MA
44. Sclaroff S, Cascia ML, Taycher L, Sethi S (1999) Unifying textual and visual cues for content-based image retrieval on the World Wide Web. Comput Vision Image Understanding 75(1/2):86–98
45. Su Z, Li S, Zhang H (2001) Extraction of feature subspaces for content-based retrieval using relevance feedback. In: ACM Multimedia'2001, Ottawa, Ontario, Canada
46. Tieu K, Viola P (2000) Boosting image retrieval. IEEE Conf. In: Computer Vision and Pattern Recognition, South Carolina
47. Tong S, Chang E (2001) Support vector machine active learning for image retrieval. In: ACM Multimedia, Ottawa, Canada
48. Tong S, Koller D (2000) Support vector machine active learning with applications to text classification. In: International Conf. on Machine Learning
49. Tsai WH, Fu KS (1979) Error-correcting isomorphism of attributed relational graphs for pattern analysis. IEEE Trans Syst Man Cybern 9:757–768
50. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
51. Vasconcelos N, Lippman A (2000) Bayesian relevance feedback for content-based image retrieval. In: IEEE Workshop CBAIVL, South Carolina
52. Vasconcelos N, Lippman A (2000) Learning from user feedback in image retrieval. Advances in Neural Information Processing Systems. MIT Press, Cambridge, MA
53. Weber R, Böhm K, Schek H (2000) Interactive-time similarity search for large image collections using parallel VA-files. In: European Conf. on Digital Libraries, San Diego, CA
54. Wood MEJ, Campbell NW, Thomas BT (1998) Iterative refinement by relevance feedback in content-based digital image retrieval. In: ACM Multimedia, Bristol, UK
55. Worring M, Smeulders A, Santini S (2000) Interaction in content-based image retrieval: a state-of-the-art review. In: International Conf. on Visual Info. Sys., Lyon, France
56. Wu L, Faloutsos C, Sycara K, Payne T (2000) FALCON: feedback adaptive loop for content-based retrieval. In: International Conf. on Very Large Data Bases (VLDB), Cairo, Egypt
57. Wu P, Manjunath BS (2001) Adaptive nearest neighbor search for relevance feedback in large image databases. In: ACM Multimedia, Ottawa, Canada
58. Wu Y, Tian Q, Huang TS (2000) Discriminant EM algorithm with application to image retrieval. In: IEEE Conf. Computer Vision and Pattern Recognition, South Carolina
59. Xu Y, Saber E, Tekalp AM (1999) Hierarchical content description and object formation by learning. In: IEEE Workshop CBAIVL, Colorado
60. Zhou XS, Huang TS (2000) A generalized relevance feedback scheme for image retrieval. In: SPIE Int. Conf. on Internet Multimedia Management Systems, Boston, MA
61. Zhou XS, Huang TS (2000) Image retrieval: feature primitives, feature representation, and relevance feedback. In: IEEE Workshop CBAIVL, South Carolina
62. Zhou XS, Huang TS (2001) Small sample learning during multimedia retrieval using BiasMap. In: IEEE Int. Conf. Computer Vision and Pattern Recognition, Hawaii
63. Zhou XS, Huang TS (2002) Unifying keywords and visual contents in image retrieval. IEEE Multimedia 9(2):23-33
64. Zhou XS, Moghaddam B, Huang TS (2001) ICA-based probabilistic local appearance models. In: IEEE Int. Conf. on Image Processing, Greece