In review for ACM Multimedia Systems Journal, shorter version appeared in IEEE CVPR2001 Workshop on Content-based Access of Image and Video Libraries (CBAIVL)

Paper Title:

Relevance Feedback in Image Retrieval: A Comprehensive Review

Authors:

Xiang Sean Zhou*, Thomas S. Huang

Beckman Institute for Advanced Science and Technology,
University of Illinois at Urbana Champaign,

Address:

405 N Mathews Ave
Urbana, IL 61801

Emails:

{*xzhou2, huang*}*@ifp.uiuc.edu*

Abstract

We analyze the nature of the relevance feedback problem in a continuous representation space in the context of multimedia information 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.

Keywords

Relevance feedback, Content-based image retrieval, Active learning, Small sample learning

1 Introduction

Initially developed in document retrieval (Salton 1989), relevance feedback was transformed and introduced into content-based multimedia retrieval, mainly content-based image retrieval (CBIR), during early and mid 1990's (Kurita and Kato 1993; Picard et al. 1996; Rui et al. 1998). Interestingly, it appears to have attracted more attention in the new field than the previous—a variety of solutions has been proposed within a short period and it remains an active research topic. The reasons can be 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.

The need for user-in-the-loop in CBIR stems from the fact that 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*. More importantly, different users at different times may have different interpretations or intended usages for the same image, which makes off-line learning unfeasible in general. Fully automated off-line preprocessing (e.g., clustering, classification) makes sense for some specific applications with well-defined image classes. But for many others, *the* best answer does not exist and ignoring user's individuality can be as senseless as trying to determine the world's greatest color.

A straightforward way of getting user into the loop is to ask the user to tune the system parameters during retrieval process, but it is too much a burden for a common user (Rui et al. 1998). A more feasible form of interaction is to ask the user to provide feedbacks regarding the (ir)relevance of the current retrieval results. The system then learns from these training examples to achieve an improved performance next round, iteratively if the user so desires.

Relevance feedback algorithms have been shown to provide dramatic performance boost in

retrieval systems (MacArthur et al. 2000; Picard et al. 1996; Rui et al. 1998; Tieu and Viola 2000; Vasconcelos and Lippman 2000a; Vasconcelos and Lippman 2000b; Worring et al. 2000).

2 The Characteristics of the Relevance Feedback Problem

Since the general assumption is that every user's need is different (Kurita and Kato 1993) and time varying, the database cannot adopt a fixed clustering structure; and *the total number of classes* and *the class membership* are not available before-hand 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 Section 4.4.)

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

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

Step 2. User provides 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 (Ishikawa et al. 1998; Meilhac and Nastar 1999) or novelty detection (Chen et al. 2001; Scholkopf et al. 2000) problem; while with both positive and negative training examples it becomes a classification problem, or an on-line learning problem in a batch mode, but with the following characteristics associated with this specific application scenario:

Small sample issue. The number of training examples is small (typically <20 per round of interaction) 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 size, some existing learning machines such as support vector machines (SVM) (Vapnik 1995) cannot give stable or meaningful results (Tieu and Viola 2000; Zhou and Huang 2001), unless more training samples can be elicited from the user (Tong and Chang 2001).

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 the 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 (Duda and Hart 1973) or SVM (Vapnik 1995), treat the 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 (Zhou and Huang 2001).

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

3 The Variants of Relevance Feedback Algorithms

It is not the intention of this section to list *all* existing techniques but rather to point out the 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 thus incomparable*. The following lists some of the conceptual dimensions along which some methods greatly differ from others:

3.1 What is the user looking for?

While most of the work assume the user is looking for "a class of similar items" to the query at hand (*"category search"*), Cox et al. (Cox et al. 2000; Cox et al. 1998) 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., the positive items 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 probable of being 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 one is "closer" to a third one *consistently* in accordance with the underlying feature representations adopted by the machine. In light of this difficulty, the user modeling has to be "soft", or *probabilistic* in nature (Cox et al. 1998).

While searching a large image database for a specific target, it is expected that in general more than one round of user interaction are needed. The machine then faces the question of "how to select the best set of images for each round to ask for user feedbacks so that the total number of iterations needed to reach the target is minimal?" (Cox et al. 1998). This issue is further elaborated in the following subsection 3.5.

3.2 What to feedback, or, what is the training data?

Some algorithms assume the user will give a binary feedback for positive and negative examples (Nastar et al. 1998; Tieu and Viola 2000; Tong and Koller 2000); some only take positive examples (Chen et al. 2001; Ishikawa et al. 1998; Rui and Huang 2000); some take positive and negative examples with "degree of (ir)relevance" for each (Rui et al. 1998; Zhou and Huang 2001); some

assumes the feedback is only a "comparative judgment" instead of a definite hit or miss (Cox et al. 1998); Some uses both labeled and unlabeled data for training: Wu et al. (Wu et al. 2000b) proposed D-EM algorithm within the *transductive learning* framework and used examples from the user feedback (labeled data) as well as other data points (unlabeled data). It performs discriminant analysis inside the 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 the computation can be a concern for large datasets.

A novel form of training is "learning from layout of images" during browsing or data visualization process(Moghaddam et al. 2001; Santini and Jain 2000). The idea is to ask the user to layout images on a "table" (i.e., a 2-D space, which can be obtained using multidimensional scaling, or MDS techniques) or to manipulate an existing 2-D layout of images, according to the user's interpretation of the semantic relationships among the 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 PCA(principle component analysis) will yield a layout of the training images that is most similar to the user's layout. The weights are then applied to the test images and PCA is used to splat ("spread flat") the test images for a 2-D image layout (Moghaddam et al. 2001).

3.3 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 (Tieu and Viola 2000) 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 2-class classifiers were formulated based on Gaussian assumption for both the 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 (Freund and Schapire 1999).

As for feature representation, while most assume one feature vector per image/region as the basic representation, Vasconcelos and Lippman (Vasconcelos and Lippman 2000a) adopted Gaussian mixture model on DCT coefficients as 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.

3.4 Class distribution.

Another issue is what assumption to be imposed on the target class(es). Gaussian assumption is a common and convenient choice (Ishikawa et al. 1998; Rui and Huang 2000). Wu et al. (Wu et al. 2000a) 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 shall be compared to a Parzen window method (Meilhac and Nastar 1999), in which Parzen window density estimation was applied to capture non-linearity in the distribution of positive examples. An elegant way to deal with non-linearity is to use reproducing kernel based algorithms. A kernel based one-class SVM as density estimator for positive examples was shown in (Chen et al. 2001) to outperform the whitening transform based linear/ quadratic method. BiasMap (Zhou and Huang 2001) and SVM active learning algorithm (Tong and Chang 2001) both adopt the kernel form to cope with nonlinear distributions, with the former emphasizing the small sample issue, while the latter exploring the active learning issue, which is discussed next.

3.5 Greedy vs. cooperative user model

If we assume 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 the

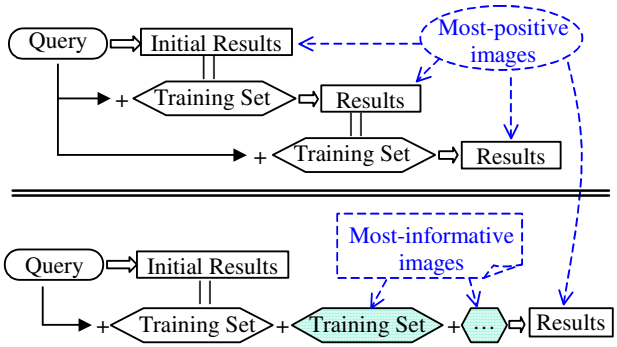

Figure 1. Top portion: the *show-me-the-results* scenario for a greedy user, where further training, if any, will be performed on the current best results;
Lower portion: the *ask-me-questions* scenario for a cooperative user, where the machine can actively select more than one screen of samples to be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

previous training. In this case, the user can terminate the query process at any point in time and will always get the best results hitherto. Additional user feedbacks or "trainings", 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 cooperative and willing to provide more than one screen of training samples before seeing the results, a new question arises: "After getting feedbacks 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 learnt."

In general the *most-informative images* (Cox et al. 2000) 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 the ones whose labels the learner is most uncertain about.

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

Active learning (Angluin 1988), or selective sampling (Freund et al. 1993), 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, in order 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 1994). Recent applications in image retrieval can be found in (Cox et al. 2000; Li et al. 2001; Tong and Chang 2001).

Cox et al(Cox et al. 2000) 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 test images.

Tong, Koller, and Chang proposed the SVM active learning algorithm for applications in text classification and image retrieval (Tong and Chang 2001; Tong and Koller 2000; Vapnik 1995). 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 the ones farthest from the boundary on the positive side in the *feature space*[2] (Vapnik 1995).

Note that the differences between (Cox et al. 2000) and (Tong and Chang 2001) 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

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

[2] A term used in kernel machine literatures to denote the new space after the nonlinear transform implied by the kernel—this shall not be confused with the *feature space* concept otherwise used to denote the representation space for descriptors extracted from the multimedia data.

extreme cases.)

Finally, there is no reason that we cannot mix the *most-informative* and *most-positive* images on one screen (Tieu and Viola 2000) —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)?

3.6 Hierarchical data organization

If a hierarchical tree structure is adopted in the database for more efficient access (Chen et al. 2000), the 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. 2000) between the speed and accuracy for searching becomes crucial. But 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 shall 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 somewhat similar to the problem of "learning the relative feature importance from a layout of images" as mentioned above in 3.2. One approach is to find the set of feature weights under which the clustering behaviors among the 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, hopeful in a way reflecting the user's under-standing and preference.

3.7 The goals and the learning machines

Along this line lies the greatest variability among different methods. A number of the early methods—self-labeled as "relevance feedback"(Chen et al. 2000; Nastar et al. 1998; Peng et al. 1999; Rui et al. 1998; Santini and Jain 2000) or not (Lowe 1995)—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

(Ishikawa et al. 1998; Rui and Huang 2000; Zhou and Huang 2001). Some of the latest work treats it either as a density estimation (Chen et al. 2001; Wu et al. 2000a), learning (MacArthur et al. 2000; Tieu and Viola 2000; Tong and Chang 2001; Tong and Koller 2000), or classification (Vasconcelos and Lippman 2000a; Wu et al. 2000b) problem.

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 (Salton 1989). These methods proposed heuristic formula-tion with empirical parameter adjustment, mainly along the line of independent axis weighting in the feature space (Peng et al. 1999; Picard et al. 1996; Porkaew et al. 1999; Rui et al. 1998; Santini and Jain 2000). The intuition is to emphasize more on the feature(s) that best clusters the positive examples and separates the positive and the negative.

Early works (Picard et al. 1996; Rui et al. 1998) had clear birthmarks from document retrieval field. For example, in (Rui et al. 1998), learning based on "term frequency" and "inverse document frequency" in text domain was transformed into learning based on the ranks of the positive and negative images along each feature axis in the continuous feature space. (Picard et al. 1996) quantized the features and then grouped the images or regions into hierarchical trees whose nodes were con-structed through single-link clustering. Groupings were then weighted using set operations.

Kohonen's *learning vector quantization* (LVQ) algorithm (Wood et al. 1998) and tree-structured *self-organizing map* (TS-SOM) (Laaksonen et al. 1999) were used for dynamic data clustering during relevance feedback. Laaksonen et al. (Laaksonen et al. 1999) used TS-SOMs to index the 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. 1999), 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. 1998) and (Rui and Huang 2000), 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 (Rui and Huang 2000) adopted a two-level weighting scheme to better cope with singularity issue due to the small number of training samples. To take into account the negative examples, Schettini et al. (Schettini et al. 1999) updated the 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. (MacArthur et al. 2000) 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 the nearest neighbors of the query were returned.

Some of the approaches are intended for off-line learning but have the potential for on-line implementation. E.g., Guo and Zhang used AdaBoost for face recognition and audio retrieval. In (Guo et al. 2001), a constrained majority voting (CMV) strategy is proposed to speed up the 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, Saber, and Tekalp (Xu et al. 1999) 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. (Ratan et al. 1999) used *multiple-instant learning* model to learn the most important sub-image(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 are shared by all positive images while far from all negative subimages. Along the same line is the work by (Forsyth and Fleck 1997), where the system learns a "body plan" for object. (Hong and Huang 2001) defined an object (or scene) as a contextual pattern and adopted ARG (attributed relational graph(Tsai and Fu 1979)) 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 learnt 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 feedbacks are to be considered. Tieu and Viola (Tieu and Viola 2000) used random sampling to get around the small sample issue, taking a risk of treating positive points as negative training samples. Vasconcelos and Lippman (Vasconcelos and Lippman 2000b) 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:

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

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. 1998) proposed empirical formulae to take into account the negative examples while estimating the distribution of the positive examples along each feature component. Zhou and Huang (Zhou and Huang 2000a; Zhou and Huang 2001) used the intuition that "all positive examples are alike in a way, each negative example is negative in its own way", and proposed asymmetric treatment for the positive and negative examples: they assumed that the positive examples have a compact low-dimensional support while the 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 the positive examples cluster while the negative scatters 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 (Forsyth and Fleck 1997; Hong and Huang 2001; Ratan et al. 1999; Xu et al. 1999).

4.2 Singularity issue in sample covariance matrix.

Many relevance feedback algorithms make use of the sample covariance matrix and its inverse (Ishikawa et al. 1998; Li et al. 2001; Rui et al. 1998; Schettini et al. 1999; Zhou and Huang 2001). 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. 1998) is not only mathematically unfounded, but also counterintuitive: Imagine a diagonal covariance matrix with the i^{th} diagonal element being 0;

according to the "weight by the inverse of the variance" heuristic implied in this method, this indicates that the i^{th} axis of the feature space is very expressive thus shall receive a very high weight. However, (Ishikawa et al. 1998) will put a weight of zero on the i^{th} axis.

Another proposal was to adopt a hierarchical weighting scheme (assuming a *block diagonal* matrix instead of the full covariance matrix) so that less parameters need to be estimated, and in the extreme case, just use a diagonal matrix (Rui and Huang 2000). This implies a forced independence assump-tion. Although intuitively appealing—the independence assumption between color and texture in some cases may be valid—for cases that 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

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

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^o, and then weight the vertical axis more. Additionally, invertibility is still not guaranteed even only diagonal elements are estimated.

It is known that under small sample the sample covariance matrix is statistically biased in the sense that the large eigenvalues become larger and the small 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" (Friedman 1989; Haff 1980; Zhou and Huang 2001):

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

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.

This operation, although simple and seemingly unintentional, actually compensates the

aforementioned bias (Friedman 1989; Haff 1980). Following the example above:

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

The solution in Equation(4) provides the rotation of 45^o (by V) followed by a proper weighting of the axes (by Λ).

To 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" (Ishikawa et al. 1998) is equivalent to a whitening transform followed by the Euclidean metric (Rui and Huang 2000) since the eigenvalues are also the singular values; when negative examples are considered using discriminant analysis as in (Zhou and Huang 2001), the generalized eigenvalues are not the same as the singular values, and the solution is a generalized whitening transform or discriminative whitening trans-form followed by the Euclidean metric.

4.3 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 2000).) 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 while clusters within each class. Since the class membership is not known *a priori*, one possible solution is to use the accumulated feedbacks from all the users as the training set to be fed into a multiple discriminant analysis (Cox et al. 1998; Duda and Hart 1973) 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

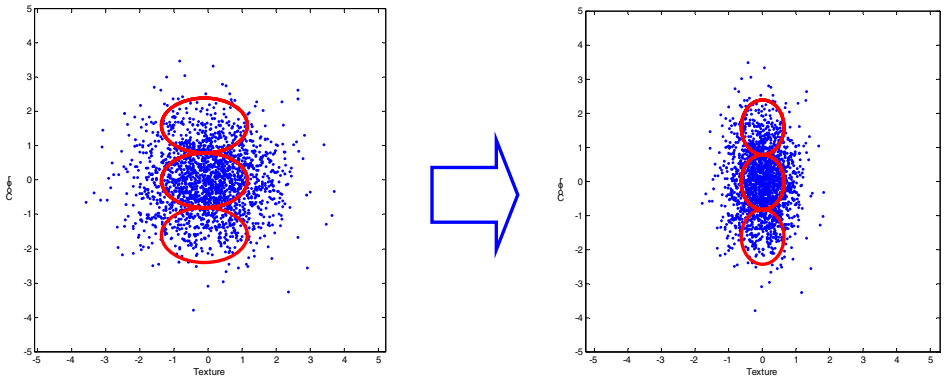

Figure 2 Feature normalization as a discriminating transformation: assuming all meaningful classes have ellipsoidal contours as shown.

assumption. A simplified example is that if all users emphasize *color feature* more than *texture* in *all* cases, there is no reason for maintaining equal variances along these two axes—stretching the *color* axis can give better initial retrieval result. This is illustrated in Figure 2. Note 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.

4.4 Pre-clustering and relevance feedback

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 off-line, automatically. But semantically meaningful clustering depends on 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 holds true for all users of the system—can be used to guide a pre-organization 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 off-line 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, as proposed by (Bartolini et al. 2001), in which the authors have assumed 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 learnt across time and across different users; and is then used to "bypass" subsequent relevance feedback loops.

Overall, in considering the pre-clustering issue, a trade-off needs to be made between flexibility (in supporting *individuality* on-line) and efficiency (by storing common knowledge off-line).

4.5 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 (Zhou and Huang 2000b). Vasconcelos and Lippman used Bayesian inference (Vasconcelos and Lippman 2000b) on image local features for relevance feedback learning. This scheme is inherently capable of image regional query, given properly learnt priors. Explicit local object learning and modeling schemes (Forsyth and Fleck 1997; Hong and Huang 2001; Ratan et al. 1999; Xu et al. 1999), if robust and fast enough, could be the ultimate choice to achieve the highest performance for image regional query through relevance feedback.

4.6 Complexity of the 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 (Wu and Manjunath 2001), 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 up the nearest neighbor search, e.g., (Weber et al. 2000).

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

5 Summery

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 the one 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 exceptions from other areas but having the essence of, or the potential of being used as, a relevance feedback algorithm for 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 have tried to point out these differences and to compare their merits. Through the comparison and analysis of the 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 one's effort in designing fast and effective relevance feedback algorithms.

Some future research directions were proposed throughout the discussion.

6 Acknowledgement

This work was supported in part by NSF Grant CDA 96-24396. Comments and suggestions from the reviewers will be greatly appreciated!

7 References

Aksoy S and Haralick RM (2000). Probabilistic vs. geometric similarity measure for image retrieval. IEEE Conf. Computer Vision and Pattern Recognition, South Carolina.

Angluin D (1988). Queries and concept learning. Machine Learning 2(3): 319-42.

Bartolini I, Ciaccia P and Waas F (2001). FeedbackBypass: A new approach to interactive similarity query processing. Int'l Conf. on Very Large Data Bases (VLDB), Rome, Italy.

Chen J-Y, Bouman CA and Dalton J (2000). Hierarchical browsing and search of large image databases. IEEE Trans. on Image Processing 9(3): 442-445.

Chen Y, Zhou XS and Huang TS (2001). One-class SVM for learning in image retrieval. Int'l Conf on Image Processing, Greece.

Cox IJ, Miller M, Minka TP, Papathomas T and Yianilos P (2000). The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. IEEE Trans. On Image Processing 9(1): 20-37.

Cox IJ, Miller M, Minka TP and Yianilos P (1998). An optimized interaction strategy for Bayesian relevance feedback. IEEE Conf. Computer Vision and Pattern Recognition, Santa Barbara, CA.

Duda RO and Hart PE (1973). Pattern Classification and Scene Analysis. New York, John Wiley & Sons, Inc.

Forsyth DA and Fleck MM (1997). Finding people and animals by guided assembly. Int'l Conf on Image Processing, Santa Barbara, CA.

Freund Y and Schapire RE (1999). A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence 14(5): 771-780.

Freund Y, Seung HS, Shamir E and Tishby N (1993). Selective sampling using the query by committee algorithm. Advances in Neural Information Processing Systems, Cambridge, MA, MIT Press.

Friedman J (1989). Regularized discriminant analysis. Journal of American Statistics Association 84(405): 165-175.

Guo G, Zhang HJ and Li SZ (2001). Boosting for content-based audio classification and retrieval: an evaluation.Microsoft Research Technical Report: MSR-TR-2001-15.

Haff LR (1980). Empirical Bayes estimation of multivariate normal covariance matrix. Annals of Statistics 8: 586-597.

Hong P and Huang TS (2001). Spatial pattern discovering by learning the isomorphic sub-graph from multiple attributed relation graphs. 8th Int'l Workshop on Combinatorial Image Analysis, PA.

Ishikawa Y, Subramanya R and Faloutsos C (1998). MindReader: query databases through multiple examples. Int'l Conf. on Very Large Data Bases (VLDB), NY.

Kurita T and Kato T (1993). Learning of personal visual impression for image database systems. Int. Conf. Document Analysis and Recog.

Laaksonen J, Koskela M and Oja E (1999). PicSOM: Self-organizing maps for content-based image retrieval. INNS-IEEE International Joint Conference on Neural Networks, Washington, DC.

Lewis DD and Gale WA (1994). A sequential algorithm for training text classifiers. ACM-SIGIR Conf. R&D in Information Retrieval, Dublin, Ireland.

Li B, Chang E and Li C (2001). Learning Image Query Concepts via Intelligent Sampling. Int'l Conf. on Multimedia and Exposition, Tokyo, Japan.

Lowe D (1995). Similarity metric learning for a variable-kernel classifier. Neural Computation 7(1): 72-85.

MacArthur SD, Brodley CE and Shyu C (2000). Relevance feedback decision trees in content-based image retrieval. IEEE Workshop CBAIVL, South Carolina.

Meilhac C and Nastar C (1999). Relevance feedback and category search in image databases. IEEE Int'l Conf. on Multimedia Comp. and Sys., Italy.

Moghaddam B, Tian Q, Lesh N, Shen C and Huang TS (2001). Visualization and layout for personal photo libraries. Int'l Workshop on Content-based Multimedia Indexing, Italy.

Nastar C, Mitschke M and Meilhac C (1998). Efficient query refinement for image retrieval. IEEE Conf. Computer Vision and Pattern Recognition, CA.

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

Picard RW, Minka TP and Szummer M (1996). Modeling user subjectivity in image libraries", in Proc. , , . Int'l Conf on Image Processing, Lausanne.

Porkaew K, Mehrotra S and Ortega M (1999). Query reformulation for content based multimedia retrieval in MARS. IEEE Int'l Conf. Multimedia Computing and Systems.

Ratan AL, O. M, Grimson W and Lozano-Perez T (1999). A framework for learning query concepts in image classification. IEEE Conf. Computer Vision and Pattern Recognition, CO.

Rui Y and Huang TS (2000). Optimizing learning in image retrieval. IEEE Conf. Computer Vision and Pattern Recognition, South Carolina.

Rui Y, Huang TS, Ortega M and Mehrotra S (1998). Relevance feedback: A power tool in interactive content-based image retrieval. IEEE Tran. Circuits and Systems for Video Tech. 8(5): 644-655.

Salton G (1989). Automatic text processing. Reading, Mass., Addison-Wesley.

Santini S and Jain R (2000). Integrated browsing and querying for image database. IEEE Trans. Multimedia 7(3).

Schettini R, Ciocca G and Gagliardi I (1999). Content-based color image retrieval with relevance feedback. Int'l Conf on Image Processing, Kobe, Japan.

Scholkopf B, Williamson R, Smola A, Shawe-Taylor J and Platt J (2000). Support vector method for novelty detection. Adv. in Neural Information Processing Systems, MIT Press.

Tieu K and Viola P (2000). Boosting image retrieval. IEEE Conf. Computer Vision and Pattern Recognition, South Carolina.

Tong S and Chang E (2001). Support vector machine sctive learning for image retrieval. ACM Multimedia, Ottawa, Canada.

Tong S and Koller D (2000). Support vector machine active learning with applications to text classification. Int'l Conf. on Machine Learning.

Tsai WH and Fu KS (1979). Error-correcting isomorphism of attributed relational graphs for pattern analysis," , vol. 9, pp. 1979. IEEE Trans. System, Man, and Cybernetics 9: 757-768.

Vapnik V (1995). The nature of statistical learning theory. New York, Springer.

Vasconcelos N and Lippman A (2000a). Bayesian relevance feedback for content-based image retrieval. IEEE Workshop CBAIVL, South Carolina.

Vasconcelos N and Lippman A (2000b). Learning from user feedback in image retrieval. Adv. in Neural Information Processing Systems, MIT Press.

Weber R, Böhm K and Schek H (2000). Interactive-time similarity search for large image collections using parallel VA-files. European Conf. on Digital Libraries, San Diego, CA.

Wood MEJ, Campbell NW and Thomas BT (1998). Iterative refinement by relevance feedback in content-based digital image retrieval. ACM Multimedia, Bristol, UK.

Worring M, Smeulders A and Santini S (2000). Interaction in content-based image retrieval: a state-of-the-art review. Int'l Conf. on Visual Info. Sys., Lyon, France.

Wu L, Faloutsos C, Sycara K and Payne T (2000a). FALCON: feedback adaptive loop for content-based retrieval. Int'l Conf. on Very Large Data Bases (VLDB), Kairo, Egypt.

Wu P and Manjunath BS (2001). Adaptive nearest neighbor search for relevance feedback in large image databases. ACM Multimedia, Ottawa, Canada.

Wu Y, Tian Q and Huang TS (2000b). Discriminant EM algorithm with application to image retrieval. IEEE Conf. Computer Vision and Pattern Recognition, South Carolina.

Xu Y, Saber E and Tekalp AM (1999). Hierarchical content description and object formation by learning. IEEE Workshop CBAIVL, Colorado.

Zhou XS and Huang TS (2000a). A generalized relevance feedback scheme for image retrieval. SPIE Int'l Conf. on Internet Multimedia Management Systems, Boston, MA.

Zhou XS and Huang TS (2000b). Image retrieval: feature primitives, feature representation, and relevance feedback. IEEE Workshop CBAIVL, South Carolina.

Zhou XS and Huang TS (2001). Small sample learning during multimedia retrieval using BiasMap. IEEE Conf. Computer Vision and Pattern Recognition, Hawaii.