

# Learning in Content-Based Image Retrieval

Thomas S. Huang<sup>1</sup>, Xiang Sean Zhou<sup>1</sup>, Munehiro Nakazato<sup>1</sup>, Ying Wu<sup>2</sup>, and Ira Cohen<sup>1</sup>

<sup>1</sup>*Beckman Institute for Advanced Science and Technology*

*University of Illinois at Urbana Champaign*

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

<sup>2</sup>*Department of Electrical and Computer Engineering*

*Northwestern University*

*yingwu@ece.northwestern.edu*

## Abstract

*In this paper we address several aspects of the learning problem in content-based image retrieval (CBIR). First, we introduce the linear and kernel-based biased discriminant analysis, or BiasMap, to fit the unique nature of relevance feedback as a small sample biased classification problem. Secondly, a WARF (word association via relevance feedback) formula is presented for learning keyword relations during the process of relevance feedback. We also introduce our new user interface for CBIR, ImageGrouper, which is designed to support more sophisticated user feedbacks and annotations. Finally, we use the D-EM (Discriminant-EM) algorithm as a way of exploiting unlabeled data in CBIR and offer some insights as to when unlabeled data will help.*

## 1. Introduction

In content-based image retrieval (CBIR), a gap exists between the high level semantics in the human minds (whether expressible by words or not) and the low level features computable by machines, even if we assume that consensus interpretation of images can be reached among all users at all times (which is seldom true). Therefore the first challenge for the machine is to *learn the associations between high-level and low-level* to effectively facilitate retrieval. On the other hand, “a picture is worth a thousand words”—a more profound challenge comes from the dynamic interpretation of images under different circumstances. In other words, the perceptual “similarity” depends upon the application, the person, and the context of usage. So in fact the machine not only needs to learn the associations, but also has to learn them *on-line with user in the loop*.

A natural way of getting user in the loop is to ask the user to give feedbacks regarding the current output of the system. Though this is an idea borrowed from the text retrieval field [1], it seems to work better in the image domain: it is easier to tell the relevance of an image than that of a document—it takes time to read through a document while an image reveals its content instantly.

Various learning techniques have been shown to provide dramatic performance boost in CBIR systems. For a detailed survey see [2].

Aside from the real-time demand, a major difficulty for learning during relevance feedback is the relatively small number of training samples available from the user, usually only 10 to 20 per round for an image retrieval system. Oftentimes this is even smaller than the dimension of the feature space. In this paper, we first present the BiasMap approach [3] to specifically address the small sample issue. We also try to model nonlinear distributions using a kernel approach. The machine learns a new query and a discriminative transform of the feature space before nearest neighbor retrieval is performed in the new space.

Some of today’s image databases have limited keyword or key-phrase annotations. It is desirable to seamlessly incorporate low-level visual feature-based learning with the semantic level keyword annotations of the image. The benefit of such integration is two-fold: we can learn not only the low-level feature support for the semantic labels [4] but also the high-level relationship between labels or keywords from the user interaction. In this paper, we also discuss a scheme for learning word association matrix during relevance feedback process [5].

For real-time learning in CBIR, user interface design plays an important role in capturing sophisticated user intentions and interpretations. Aside from providing judgments on relevance or irrelevance, the user might wish to select and group examples from multiple rounds of interactions, or make overlapping annotations to different groups of images. Such actions can be supported by *ImageGrouper*, a novel interface we proposed for CBIR [6].

The basic setup in content-based image retrieval is usually as a binary classification problem—one class for the query image class, and the other for the rest. Learning these classifiers is usually based on labeled records only, obtained through feedback from the user on relevant images and irrelevant images. The rest of the images in the database, which can be considered unlabeled data, are not utilized in building the classifier. The question is whether they can be used to enhance the performance of the retrieval system.

In the past few years there has been a growing interest in the use of unlabeled data for enhancing classification accuracy in supervised learning settings. Most studies on the subject have shown that there is great potential in using unlabeled data to enhance the learning process and improve classification results [7]. We have proposed D-EM (Discriminant-EM) algorithm [8] to incorporate unlabeled data in a semi-supervised fashion for enhancing the performance in content based image retrieval.

However it is easy to foresee situations, where the unlabelled data will harm the classifier. Some of our recent work explores the conditions under which unlabelled data can be helpful [9, 10].

For applications where a user annotation step is performed, the question arises as to how to relieve the user of the burden of labeling the whole dataset. One solution is to explore the clustering structure of the data and select representative points proactively to query the user, and then propagate the labels automatically [11]. This idea is closely related to *active learning*.

Active learning [12], or selective sampling [13], studies the strategy for the learner (i.e., the machine) to actively select samples to query 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 [14]. Recent applications in image retrieval can be found in [15, 16, 17].

The principles of incremental learning can certainly be applied in interactive image retrieval. A relevance feedback process is inherently incremental: except for the first round, the machine always learns from both previous feedbacks from the user(s), and the new information from the current round. Incremental knowledge accumulation can be performed either for one user only or among different users and across different feedback sessions, depending upon whether the underlying clustering structure in one user's mind differs from those in the others'. [18, 2]

More recent work also explores the learning issue related to image visualization or layout [19, 20]. Assuming the user lays out photos on a table (or the computer screen) in a certain way, the machine can learn a feature space transformation to match the mutual distances among these images (e.g., using multi-dimensional scaling techniques). The learned feature transform, when applied on test images, shall place them in a way mimicking the user's preferences.

We shall point out that all the aforementioned learning techniques are based on the premise that the image representation in the feature space has retained the high-level semantic information necessary to support the user's interpretation. Unfortunately, this requirement is often not satisfied.

In the following, we discuss some of the work done at our group in the area related to learning in CBIR.

## 2. Small Sample Learning by Relevance Feedback

Much work regards the relevance feedback problem as a strict two-class classification problem, with equal treatments on both positive and negative examples [2]. It is reasonable to assume positive examples to cluster in certain way (maybe non-linearly), but negative examples usually do not cluster since they can belong to any class. Forcefully assigning all negative examples into one class/mode can mislead the algorithm therefore hurt the robustness in performance, especially when the number of training samples is small.

We define *biased classification problem* as the problem in which there are an unknown number of classes but the user is only interested in one class, i.e., the user is *biased toward one class*. And the training examples are labeled by the user as only positive or negative as to whether they belong to the target class or not. Thus the negative examples can come from an uncertain number of classes. More importantly, the limited number of negative examples is not representative of the true distribution of all negative points. (One alternative is to use a random sampling strategy to increase the number of negative examples thus their representative power. But this carries the danger of mislabeling.) Therefore it is desirable to distinguish a real two-class problem from a biased classification (or  $(1+x)$ -class) problem in our case. And a typical relevance feedback process poses a biased classification problem.

For a biased classification problem, we ask the following question: what is the optimal linear transformation to take into account both positive and negative examples, such that positive examples have minimal scatter while negative examples have maximal scatter with respect to positive ones (i.e., negative examples are "far away" from positive ones)? Or mathematically, what is the optimal transformation such that the ratio of "positive scatter" over "negative scatter with respect to positive" is minimized? We call this biased discriminant analysis (BDA) due to the bias toward the positive examples.

Linear/nonlinear BDA, which we dub it BiasMap, is a supervised learning scheme using a small number of positive and negative examples. It is a mapping or transformation of the representation space, usually into a lower dimensional subspace. It is *biased* in the sense that positive and negative examples are treated asymmetrically.

### 2.1 Biased Discriminant Analysis (BDA)

We use  $\{x_i\}$ ,  $i=1,\dots,N_p$  to denote the positive examples, and  $\{y_i\}$ ,  $i=1,\dots,N_N$  to denote the negative examples. And  $m_x$  is the mean vector of  $\{x_i\}$ . For any linear transformation matrix  $W$ , which is a long matrix, we define the biased criterion function

$$J(W) = \frac{|W^T S_N W|}{|W^T S_P W|} \quad (1)$$

where

$$S_N = \sum_{i=1}^{N_N} (y_i - m_x)(y_i - m_x)^T \quad (2)$$

and

$$S_P = \sum_{i=1}^{N_P} (x_i - m_x)(x_i - m_x)^T \quad (3)$$

The optimal biased linear transformation is

$$W^o = \arg \max_w \{J(W)\} \quad (4)$$

The difference from traditional discriminant analysis formulation lies in the asymmetry.

To avoid singularity in estimating the covariance matrix in the solution, regularization shall be applied by adding small positive values to the diagonal before inversion. The influence of the negative examples can be tuned by adding a discounting factor, i.e., a positive value to the diagonal—with a large discounting factor BDA is reduced to the whitening transform on positive examples.

## 2.2 Nonlinear BDA using Kernel

To take into account nonlinearity in the data distributions, we propose a kernel-based approach [21, 22]. The original BDA problem is reformulated into dot-product form, and then a kernel is used in place of the dot product. This is the major advantage of kernel-based approach—if we were to carry out the nonlinear transformation explicitly before the dot-product calculation, the computation would usually be formidable. Assume a nonlinear transform  $\Phi$  on the original data before applying linear BDA:

$$J(W) = \frac{w^T S_y^\phi w}{w^T S_x^\phi w} \quad (5)$$

It can be shown that the numerator of (5) can be rewritten as:

$$w^T S_y^\phi w = \alpha^T (K_y - K_x I_{N_x}^y) (K_y - K_x I_{N_x}^y)^T \alpha \quad (6)$$

where:

$$K_{y_j} = \Phi^T \phi(y_j), \quad K_{m_x} = \Phi^T m_x^\phi, \quad (K_y)_{:,j} = K_{y_j} \quad (7)$$

and  $I_{N_x}^y$  is an  $N_x$  by  $N_y$  matrix of all elements being  $1/N_x$ .

Similarly, rewrite the denominator of (5),

$$w^T S_x^\phi w = \alpha^T K_x (I - I_{N_x}^x) K_x^T \alpha \quad (8)$$

and  $I_{N_x}^x$  is an  $N_x$  by  $N_x$  matrix of all elements being  $1/N_x$ .

With optimal  $\alpha$ , the projection of a new pattern  $z$  onto  $w$  is given by:

$$w^T \phi(z) = \sum_{i=1}^{N_x} \alpha_i k(x_i, z) + \sum_{j=1}^{N_y} \alpha_{j+N_x} k(y_j, z) \quad (9)$$

Extensive experiments have demonstrated the advantages of the proposed algorithm. Here we can only report one experiment on a fully labeled set of 500 images from COREL. It contains five classes, each with 100 images. Visual features used are color moments, wavelet moments, and water-filling structural features [23]. Each round 10 positive and 10 negative images are randomly drawn as training samples and the learned feature transformation is applied to all the images before a nearest neighbor classifier is applied for image ranking. For each round the error rate in the top 100 returns is recorded as the performance measures. A total of 500 rounds of testing are performed and the averaged error rates are shown in Table 1, where four schemes are compared: QM (Query Movement, i.e., no transformations, just return the nearest neighbors of the positive centroid), WT (whitening transform on positive examples [24]), BDA, and KBDA. Evident performance improvement is observed.

**Table 1.** Averaged hit rate in top 100 for 500 trials

QM	WT	BDA	KBDA
62.7%	70.4%	74.2%	79.1%

## 3. Learning Word Association via Relevance Feedback

The performance of a content-based image retrieval (CBIR) system is inherently constrained by the use of the low-level features, and can not give satisfactory retrieval results in many cases; especially when the high-level concepts in the user's mind is not easily expressible in terms of the low-level features. Therefore, for real world applications, whenever possible, textual annotations shall be added or extracted and processed to improve the retrieval performance. We have recently explored the unification of keywords and feature contents for image retrieval. We proposed a seamless joint querying and relevance feedback scheme based on both keywords and low-level visual contents incorporating keyword similarities. We proposed an algorithm for the learning of the word similarity matrix during user interaction, namely word association via relevance feedback, or WARF. This learned similarity matrix, specific to the dataset as well as the users, can be applied for keyword semantic grouping, thesaurus construction, and soft query expansion during intelligent image retrieval with user-in-the-loop [25, 5].

The assumption is that some of the images in the database have textual annotations in terms of short phrases or keywords. These can come from pattern recognition; automatic speech recognition, keywords spotting from text (e.g., from surrounding HTML text on web-pages); or manual annotation, etc.

Among all the images shown to the user, the *relevant set* is the set of images indicated by the user as “good” ones. If a term only appeared in the annotations for the images in the *relevant set*, it is called a *relevant term*.



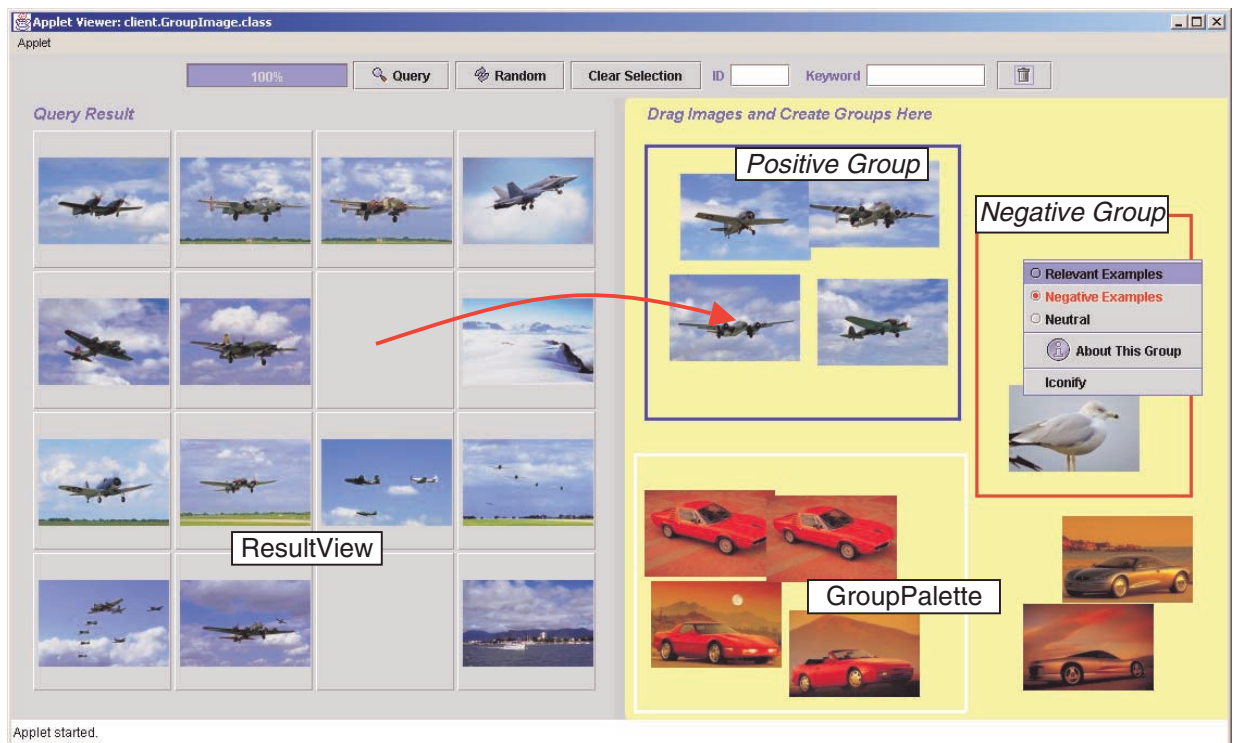


Figure 1. *ImageGrouper* interface

The number of occurrences of a *relevant term*  $i$  in the *relevant set* is called *relevant term frequency*, denoted as  $f_i$ . The number of co-occurrences of two relevant terms  $i$  and  $j$  in the same image is denoted by  $c_{ij}$ . The “relevance” of term  $i$  and  $j$ ,  $S_{ij}$ , is then updated as

$$S_{ij} = S_{ij} + \max(f_i, f_j) \times (\min(f_i, f_j) - c_{ij}) \quad (10)$$

This formula for updating word association through relevance feedback, or the WARF formula, is executed after every user feedback with more than one relevant image.

The result of multiple WARF updates is used to facilitate the unification of keywords and contents in a flexible and meaningful way. During each user retrieval and feedback process, the algorithm will dynamically update the weights in a semantic network consisting of the keywords appeared in the database. This algorithm was tested through simulated experiments [25].

After extensive use of the database by the user, the output of the algorithm, i.e., the weights between pairs of terms will correspond to the “similarity” of the two terms, or the estimated probability for the user to request these two terms together in one query. By using Hopfield network or clique detection, we can further group terms into semantic classes, which can be used to assist future retrieval process.

In addition, since these “knowledge” are all extracted from the user feedback, the word association information can also be regarded as the user’s “search habit” or “preference”. Therefore, this real time thesaurus construction algorithm based on user feedback will provide

a practical way for not only the semantic grouping of keywords but also learning of user preference.

#### 4. User Interface Support for Relevance Feedback

While relevance feedback for image retrieval was very successful, there are several drawbacks in the traditional user interfaces. First, these systems assume that *the more query examples are available, the better result we can get*. Therefore, only way to refine the search was to add example images incrementally. However, this assumption is not necessarily true. There are cases where additional examples may contain undesired features and degenerate the retrieval performance. [6] showed finding good combinations of query examples is essential for successful image retrieval. Second, the traditional interface does not allow the user to put aside the query results for later uses. This type of interaction is desired because the users of CBIR are not necessarily looking for only one type of images. The users’ interest may change during retrieval. In addition, there was no way to integrate content-based retrieval with keyword annotation. To overcome these limitations, we are developing a new user interface for image retrieval, named *ImageGrouper* [6].

Figure 1 shows the interface of *ImageGrouper*. The interface is divided into two panes. The left pane is the *ResultView* that displays the result of both content-

based and keyword-based search. The right pane is the *GroupPalette*, in which the user handles image groups. In order to create an image group, the user first drags images from the *ResultView* to *GroupPalette*, then encloses the images by drawing a box as we draw rectangles in an illustrating application. The user can create any number of groups within the palette. For each group, the user can specify it as relevant, irrelevant or neutral. Neutral groups do not contribute to the search at the moment and can be used for another query later. The properties of the groups can be easily changed from the popup menu. The box colors change accordingly. Moreover, the user can move images from one group to another at any moment. In addition, these groups can be overlapped to each other, allowing images to belong multiple groups. When the user presses the “Query” button, the system retrieves images that are similar to positive group images while avoiding images similar to negative groups images. The results are displayed in the *ResultView*. The user can refine the query further by dragging the new images into groups in the palette.

Compared with the traditional GUI for CBIR, *ImageGrouper* is much easier to try different combinations of query examples. Moreover, it provides storage area for images that are not used for query at the moment. It is shown that this *trial and error* query can improve the query performance even if underlying algorithms are identical [6].

In addition, the user can create new groups within an existing group (Group in a group). Thus, the user can begin with collecting relatively generic images first, then narrow down the search to more specific images. For example, assume the user is looking for “red car” images. S/he can initially collect “cars with any colors.” Once s/he found enough number of images, s/he can divide them into two groups: red cars and other cars. Then the user can specify red car group as relevant, and other cars as irrelevant. This narrowing down search was not possible with the conventional interfaces.

Furthermore, the user can annotate text information to each group that is obtained by content-based queries. This text information can be used for keyword-based search. Therefore, *ImageGrouper* makes it possible integration of content-based and keyword-based query. The user first creates groups of images using content-based query, then s/he annotate these groups with keywords. S/he can retrieve the images with keywords later.

## 5. Using Unlabeled Data

The retrieval system acts as a classifier to divide the images in the database into two classes, either relevant or irrelevant. In this sense, an annotated image can be represented by a feature vector  $\mathbf{x}$ , e.g. a set of image features or eigen features, and its label  $y$  that is either

relevant or irrelevant, i.e.,  $y \in \{+1, -1\}$ . It seems that many supervised learning approaches could be employed to approach to this classification problem. Unfortunately, they are confronted by two main challenges. The first one is that the annotated or labeled training samples are too limited. Generally, the labels are provided by queries and relevance feedbacks, which will not be many. Limited training data would only result in weak classification. Another challenge is the dimensionality of learning, since high dimensional visual data would pose practical difficulties for feature weighting, selecting and dimensionality reduction. Limited training data would also prevent effective dimensionality reduction schemes. Therefore, a new learning scheme needed for such a challenging scenario.

### 5.1 D-EM Algorithm

Considering there are a large number of unlabeled images in the given database, we may use them to boost the weak classifier learned from the limited labeled data, since unlabeled data contain information about the joint distribution over features. In such circumstance, the hybrid training data set  $D$  consists of a labeled data set  $L = \{(\mathbf{x}_i, y_i), i=1, \dots, N\}$ , and an unlabeled data set  $U = \{\mathbf{x}_i, i=1, \dots, M\}$ . In CBIR, the query images act as the labeled data, and the whole database or a subset can be treated as the unlabeled set. We make an assumption here that  $L$  and  $U$  are from the same distribution. Therefore, image retrieval is to classify the images in the database based on both labeled and unlabeled training data.

The Expectation-Maximization (EM) approach can be applied to this learning task, since the labels of unlabeled data can be treated as missing values. A parameterized generative model can be used to model the data distributions. For example, we can use Gaussian mixture models. The E-step basically estimates the expected membership for each unlabeled sample to augment the labeled training set, and the M-step estimate parameters of the generative model based on such an augmented data set. The iteration between the E-step and the M-step is expected to improve the classifier to a local optimum in an incremental sense. Unfortunately, in practice, such a standard EM approach will be plagued by some difficulties. One is that our assumption in the generative model on the probabilistic structure of data distribution, e.g., the number of mixture components, may not hold, since we may not know the ground truth data distributions in advance. Such a discrepancy may diverge the EM iteration. Secondly, a large number of parameters need to be estimated due to the high dimensionality of the generative model. Consequently, the standard EM approach often fails when such a structure assumption does not hold.

To alleviate such difficulties, we can find a mapping such that the data are clustered in the mapped feature

space, in which the probabilistic structure could be simplified and captured by simpler Gaussian mixtures. Putting the step of finding the mapping into the EM iteration loop, we propose a three-step iteration method, the Discriminant-EM (D-EM) algorithm [8, 26, 27], in which the E-step estimates the membership for each training data, D-step identifies a mapping and thus a discriminating subspace, and M-step estimates the parameters of the generative model in the lower dimensional discriminating subspace. The advantage of putting the D-step in the loop is that the E-step has provided a large augmented labeled training set. The D-EM algorithm iterates among these three steps.

We use multiple discriminant analysis (MDA) as the D-step. The basic idea behind MDA is to find a mapping such that the ratio of the between-class scatter and within-class scatter is maximized in the projected space. MDA offers a means to catch major differences between classes and discount factors that are not related to classification. Another advantage of MDA is that the data are clustered to some extent in the projected space, which makes it easier to select the structure of Gaussian mixture models. We developed a nonlinear kernel MDA algorithm based on kernel machines [26]. Both linear MDA and kernel MDA have been used in the D-EM algorithm [8, 26, 27].

The application of D-EM to image retrieval is straightforward [8]. Based on relevance feedback, several relevant and irrelevant examples are labeled by human. Using a random subset of the database or even the whole database as an unlabeled data set, the D-EM algorithm identifies some “similar” images to the labeled images to enlarge the labeled data set. Therefore, good discriminating features could be automatically selected through this enlarged training data set to better represent the implicit concepts. In our current implementation, in the transformed space, both classes are represented by Gaussian distributions. The D-EM iteration tries to boost an initial weak classifier.

In order to give some analysis and compare several different methods, we manually labeled an image database of 134 images. Our dataset has 7 classes such as airplane, bird, car, church painting, flower, mountain view and tiger. All images in the database have been labeled as one of these classes. In all the experiments, these labels for unlabeled data are only used to calculate classification error. We fed the D-EM algorithm a different number of labeled and unlabeled samples. We found that D-EM brought about 20% to 30% more accuracy than without using unlabeled data.

## 5.2 When Will Unlabeled Data Help?

In our recent work [9, 10] we have discussed the value of unlabeled data in supervised statistical learning using generative classifiers. We present here some of the main conclusions and results of this work, which

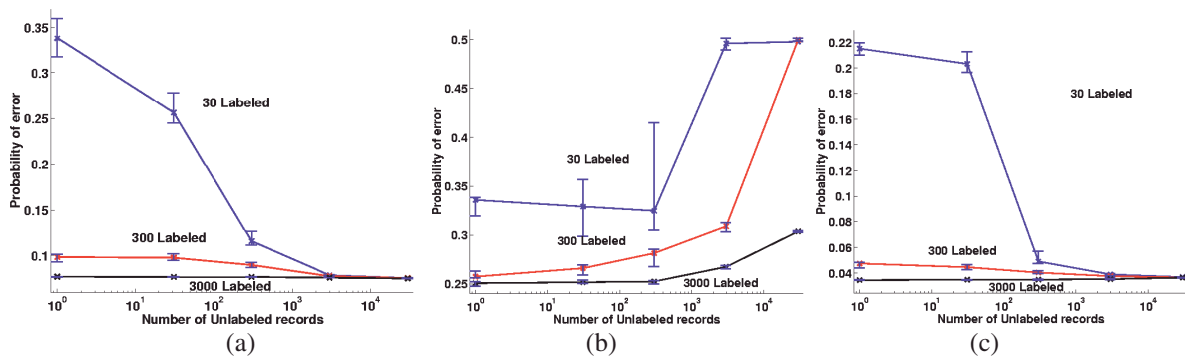
can be used as a guideline for any system which wishes to utilize unlabeled data with labeled data.

Defining the problem, let  $C$  be the class variable takes a finite number of labels, denoted as  $|C|$ . Let  $X$  be the feature vector. We are given a data set,  $D$ , of independent instances sampled from some unknown joint distribution  $P(C, X)$ .  $D$  has  $N$  labeled instances and  $M$  unlabeled instances, where an unlabeled instance is missing the value for  $C$ . Our task is to learn the parameters of the joint probability distribution  $P(C, X)$  and use it to perform classification by computing  $P(C | x)$  for a given feature vector  $x$ . One of the basic methods to estimate this distribution is maximum-likelihood (ML) estimation.

Past theoretical and experimental work showed that using the ML estimation approach (via EM or other numerical algorithms when unlabeled data is present) should yield improvement in the classification accuracy as more unlabeled data is added. However, there is an underlying assumption in all the theories - that the assumptions on the joint distribution are correct (for example, dependencies between features, the distribution of the features, whether Gaussian or other). This restriction is quite harsh. It requires good prior knowledge of the nature of the joint distribution. In practice however, such knowledge is usually not available. The question is how unlabeled data affect the classification accuracy in these cases.

Figure 2 shows three examples of learning with unlabeled data. In Figure 2(a) we show the result of adding unlabeled data to fixed size labeled data sets (30, 300, 3000) when we have the correct prior knowledge of the distribution that generated the data. As can be seen, adding unlabeled data improves greatly the classification performance, even for the case using only 30 labeled records. Figure 2(b) shows the case of learning with incorrect assumptions on the joint distribution. In this case, adding the unlabeled data decreases the performance of the classifier. As we add more unlabeled data the performance degrades more and more. This is a startling result since the common statistical folklore is that more data should help, but in this case it makes the classifier worse. Figure 2(c) shows a case of learning with incorrect assumptions on the joint distribution, but the complexity of the problem is relatively high (49 features compared to 10 features in case (a) and (b)). In this case we see that when the labeled data set is small (30 labels and 300 labels), adding the unlabeled data improves the classification performance. However, when 3000 labeled data are available, adding the unlabeled data decreases the performance.

To summarize, unlabeled data consistently improve classification performance when the correct assumptions are made on the joint distribution. When incorrect assumptions are made (which is often the case), unlabeled data can both improve and degrade the classification performance, depending on the complexity of the



**Figure 2** (a) Assumed and correct model match, (b) Assumed and correct models do not match with 10 features, (c) Assumed and correct models do not match, with 49 features. The bars represent 30% and 70% percentiles.

classifier compared to the size of the labeled training set. The results also suggest a new use for unlabeled data: detecting incorrect assumptions, since the classification performance can only degrade if the assumptions on the joint probability were incorrect.

Note that a key point of the D-EM algorithm is that the D-step *actively tunes* the data in a reduced space so that the assumptions implied by the E and M-steps can hold with a much higher probability.

## 6. Conclusion

In this paper, we have briefly introduced our work related to the learning issue in CBIR. For relevance feedback, three schemes are discussed with the first one attacking the small sample learning issue, the second attempting the integration issue of visual features and keywords, and the third addressing the user interface design. We also discussed the use of unlabeled data in CBIR, with a careful look at the conditions under which unlabeled data will help.

**Acknowledgements:** This work was supported in part by NSF Grant CDA 96-24396 and EIA 99-75019.

## Reference

[1] G. Salton, *Automatic text processing*, Reading, Mass.: Addison-Wesley. 1989.

[2] X.S. Zhou and T.S. Huang. "Exploring the nature and variants of relevance feedback," in proc. *IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL)*, in conjunction with CVPR. Hawaii. Dec. 2001.

[3] X.S. Zhou and T.S. Huang. "Small sample learning during multimedia retrieval using BiasMap," in proc. *Int'l Conf. Computer Vision and Pattern Recognition*. Hawaii. Dec. 2001.

[4] K. Barnard and D.A. Forsyth. "Learning the semantics of words and pictures," in proc. *Int'l Conf. on Computer Vision*. Vancouver. July 2001.

[5] X.S. Zhou and T.S. Huang. "Unifying keywords and contents for image retrieval," in proc. *International Workshop on Content-Based Multimedia Indexing*. Italy. Sept. 2001.

[6] M. Nakazato, L. Manola, and T.S. Huang. "ImageGroup: Search, Annotate and Organize Image by Groups," in proc. *The Fifth International Conference on Visual Information System (VIS'2002)* 2002.

[7] M. Seeger, *Learning with labeled and unlabeled data*, in *Technical Report*. 2001, Edinburgh University, UK.

[8] Y. Wu, Q. Tian, and T.S. Huang. "Discriminant EM algorithm with application to image retrieval," in proc. *IEEE Conf. Computer Vision and Pattern Recognition*. South Carolina. June 2000.

[9] F.G. Cozman and I. Cohen. "Unlabeled data can degrade classification performance of generative classifiers," in proc. *The 15th International FLAIRS Conference* 2002.

[10] I. Cohen, F.G. Cozman, and A. Bronstein. "The effect of unlabeled data on generative classifiers, with application to model selection," in proc. *AAAI (Submitted)* 2002.

[11] M.R. Naphade, X.S. Zhou, and T.S. Huang. "Image classification using labeled and unlabeled images," in proc. *SPIE Intl. Symposium on Voice, Video and Data Communications, Program on "Internet Multimedia Management Systems"*. Boston, MA. Nov 2000.

[12] D. Angluin, Queries and concept learning. *Machine Learning*. 2.(3)1988 p. 319-42.

[13] Y. Freund, et al. "Selective sampling using the query by committee algorithm," in proc. *Advances in Neural Information Processing Systems*. Cambridge, MA. MIT Press 1993.

[14] D.D. Lewis and W.A. Gale. "A sequential algorithm for training text classifiers," in proc. *ACM-SIGIR Conf. R&D in Information Retrieval*. Dublin, Ireland. July 1994.

[15] I.J. Cox, et al., The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. *IEEE Trans. On Image Processing*. 9.(1)2000 p. 20-37.

[16] B. Li, E. Chang, and C. Li. "Learning Image Query Concepts via Intelligent Sampling," in proc. *Int'l Conf. on Multimedia and Exposition*. Tokyo, Japan. August 2001.



- [17] S. Tong and E. Chang. "Support vector machine active learning for image retrieval," in proc. *ACM Multimedia*. Ottawa, Canada. Sept. 2001.
- [18] Z. Su, S. Li, and H. Zhang. "Extraction of feature subspaces for content-based retrieval using relevance feedback," in proc. *ACM Multimedia'2001*. Ottawa, Ontario, Canada. Sept. 30-Oct 5 2001.
- [19] B. Moghaddam, Q. Tian, and X.S. Zhou. "Visualizing images using extracted features," in proc. *US Patent. pending*. 2001.
- [20] B. Moghaddam, et al. "Visualization and layout for personal photo libraries," in proc. *Int'l Workshop on Content-based Multimedia Indexing*. Italy. Sept. 19-21 2001.
- [21] V. Vapnik, *The nature of statistical learning theory*, New York: Springer. 1995.
- [22] S. Minka, et al. "Fisher discriminant analysis with kernels," in proc. *IEEE workshop on Neural Networks for Signal Processing* 1999.
- [23] X.S. Zhou and T.S. Huang, Edge-based structural features for content-based image retrieval. *Pattern Recognition Letters*. 22.2001 p. 457-468.
- [24] Y. Rui and T.S. Huang. "Optimizing learning in image retrieval," in proc. *IEEE Conf. Computer Vision and Pattern Recognition*. South Carolina. June 2000.
- [25] X.S. Zhou and T.S. Huang, Unifying keywords and visual contents in image retrieval. *IEEE Multimedia*,(April-June)2002 p.
- [26] Y. Wu, K. Toyama, and T.S. Huang. "Self-supervised learning for object recognition based on kernel discriminant-EM algorithm," in proc. *IEEE Int'l Conference on Computer Vision*. Vancouver. July 2001.
- [27] Y. Wu and T.S. Huang. "Towards self-exploring discriminating features," in proc. *IAPR Int'l Workshop on Machine Learning and Data Mining in Pattern Recognition (MLDM'01)*. Leipzig, Germany. July 2001.