Small Sample Learning during Multimedia Retrieval using BiasMap

Xiang Sean Zhou, Thomas S. Huang
Beckman Institute, University of Illinois at Urbana Champaign
{xzhou2, huang}@ifp.uiuc.edu

Abstract

All positive examples are alike; each negative example is negative in its own way.

During interactive multimedia information retrieval, the number of training samples fed-back by the user is usually small; furthermore, they are not representative for the true distributions—especially the negative examples. Adding to the difficulties is the nonlinearity in real-world distributions. Existing solutions fail to address these problems in a principled way. This paper proposes biased discriminant analysis and transforms specifically designed to address the asymmetry between the positive and negative examples, and to trade off generalization for robustness under a small training sample. The kernel version, namely "BiasMap", is derived to facilitate non-linear biased discrimination. Extensive experiments are carried out for performance evaluation as compared to the state-of-the-art methods.

1. Introduction

To design a content-based multimedia information retrieval system, one needs to address at least two issues: the first is how to find effective and compact representations for the data; the second is how to select distance metrics for data ranking in accordance with the human perception of the contents. The latter shall be dealt with in real-time with user in the loop because the metric *dynamically* depends upon the user and the context. This is the focus of this paper, i.e., real-time learning of distance metric, or feature space transformations based on the user interactions. For this purpose we will assume the selected feature representations are effective to the necessary extent.

The need for on-line learning stems from the fact that different semantic concept lies in different subspace, and the selection of such subspaces cannot be done off-line in general, since different users at different times often have different interpretations or requests regarding the same piece of information. And it will be a burden for a regular user to tune the internal parameters for the machine to adapt to changes of subspace. Such difficulties have led to research efforts relating to on-line learning of user preferences, interpretations, or retrieval requirements. These are referred to as *relevance feedback* algorithms [14].

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

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

Then, iteratively:

- *User provides judgment on the current results as to whether, and to what degree, they are relevant or irrelevant to her/his request;*
- *The machine learns and tries again.*

It is worth noting that there are variants of relevance feedback algorithms that have used different assumptions from those adopted in this paper. For an extensive review and comparison the reader is referred to [14].

Initiated in document retrieval field, much of the relevance feedback research has been recently conducted in the field of content-based image retrieval (CBIR). But they can be suitable for retrieval of other media types as well. In this paper, we assume each image or each unit of information is represented by a vector. In the abstraction of the feature space, each unit becomes a point. Relevance feedback with both positive and negative training examples becomes a supervised classification problem, or an on-line learning problem in a batch mode, but with some unique characteristics [14].

Most of the state-of-the-art techniques use ad hoc heuristics to deal with positive and negative feedbacks, imposing arbitrary feature independence assumptions [14]. Some techniques offer optimal solutions but only on positive examples [2], or only in a linear/Euclidean sense [6][8]. This paper proposes a novel scheme called *BiasMap* that can deal with positive and negative examples with non-linear densities asymmetrically in a principled way (Section 3). Extensive experiments and evaluations are reported in Section 4.

2. Traditional discriminant analysis

From the pattern analysis point of view, when only positive examples are to be considered and with Gaussian assumption, the whitening transformation is the optimal choice, which is equivalent to the use of Mahalanobis distance metric [3](*cf.* [6][8]). When both positive and negative examples are considered, instead of various seemingly plausible heuristics for feature-weighting [14], two optimal linear transformations based on the traditional discriminant analysis are worth discussing:

2.1 Two-class or multi-class

One is the two-class (or more specifically, "two-mode") fisher discriminant analysis (FDA). The objective is to find a subspace in which the ratio of between-class scatter over within-class scatter is maximized. (See [3] or [5] for details.) However, it is part of the objective that negative examples shall cluster in the discriminating subspace. This is an unnecessary and potentially damaging requirement since very likely the negative examples belong to multiple classes/modes; furthermore the limited number of negative examples cannot represent the true distribution well. (One alternative is to take random samples and assume all of them to be negative examples, which is not true but may hold with high probability [11].)

Another choice is to use multiple discriminant analysis (MDA), where we treat each negative example as from a different class/mode. It becomes a (N_y + 1)-class discriminant analysis problem, where N_y is the number of negative examples. In this setting, it is part of the objective that all negative examples shall scatter in the subspace. This is again an unnecessary and potentially misleading requirement since several negative examples can come from the same class. The effort to split them up will yield poor results, and the damage is most severe when we have more negative than positive examples.

2.2 More alternatives

Some may argue that unsupervised clustering techniques—EM using minimum description length criteria, or mean shift—can be applied to find out the number of clusters automatically. But meaningful clustering depends on the subspace selection—an image of a "red table" is not necessarily closer to a "white table" than a "red horse" unless a proper discriminating subspace can be specified in the first place—which is exactly what the system is trying to learn; and even if the clusters can be obtained, *any constraint put on the negative examples other than "stay away from the positive" is unnecessary and misleading*. Finally iterative clustering can be too time consuming.

Nonparametric discriminant analysis [5] can be a choice in the right direction for modeling certain non-linearity in class distributions and achieving higher *effective dimensions*. But its solution is limited to be either linear or quadratic; and it still adopts the two-class assumption and treats them equally, which is the same as FDA.

3. Biased discriminant analysis and BiasMap

Instead of confining ourselves to the traditional settings of the discriminant analysis, a better way is to use a new form of the discriminant analysis to suit our objective, for which we believe that the relevance feedback problem is better cast as a "biased learning problem":

3.1 (1+*x*)-class Assumption

(1+ *x*)-*class learning* or *biased learning* can be defined as the learning 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 samples 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. Past research has addressed this problem simply as a two-class classification problem with symmetric treatment on positive and negative examples, which makes sense only when sufficient negative examples are available. However the situation for relevance feedback during information retrieval is that the negative examples are too few to be of representative power for the true distribution. While the positive examples may have a better chance since in reality the class-of-interest usually has compact support—the intuition is that "all positive examples are alike, each negative example is negative in its own way" (*cf.* First sentence of Leo Tolstoy's *Anna Karenina*). When the negative examples are too few to be representative of their true distribution, the (1+*x*)-class assumption becomes critical.

3.2 Objective function formulation

With asymmetric treatment biased toward the positive examples, we can write the objective function as:

$$W_{opt} = \arg\max_{W} \frac{\left|W^T S_y W\right|}{\left|W^T S_x W\right|} \qquad (1)$$

Where the scatter matrix *estimates* can be obtained by

$$S_y = \sum_{i=1}^{N_y} (y_i - m_x)(y_i - m_x)^T \qquad (2)$$

$$S_x = \sum_{i=1}^{N_x} (x_i - m_x)(x_i - m_x)^T \qquad (3)$$

$\{x_i,\ i = 1, \ldots, N_x\}$ denote the positive examples, and $\{y_i,\ i = 1, \ldots, N_y\}$ are the negative examples. Each element of these sets is a vector of length n, with n being the number of feature components. m_x, m_y, and m are the mean vectors of the sets $\{x_i\}$, $\{y_i\}$, and $\{x_i\} \cup \{y_i\}$, respectively.

The aim is to find an optimal transform that clusters only positive examples while keeping negatives away.

The Gaussian assumption is imposed on positive examples in this case (See section 3.6 for nonlinear case).

The difference from existing formulae [5] is subtle, but critical. Two points are worth noting: one is the *asymmetry*; the other is the increased *effective dimensions*: due to the ranks of S_y and S_x, BDA has *effective dimension* of $\min\{N_x, N_y\}$, while for FDA, it is only 1 [5]. This gives BDA significantly higher capacity for informative density modeling, for which FDA has virtually none. This difference is partially responsible for BDA's robust performance under small sample size as compared to FDA, even in the kernel form (See section 4.2.2).

3.3 Regularization and Discounting Factors

It is well known that the sample-based plug-in estimates of the scatter matrices based on Equations (2) and (3) will be severely biased for small number of training samples, in which case regularization is necessary to avoid singularity in the matrices. This is done by adding small quantities to the diagonal of the scatter matrices [4]. The regularized version of S_x, with n being the dimension of the original feature space and I the identity matrix, is:

$$S_x^r = (1-\mu)S_x + \frac{\mu}{n} tr[S_x]I \tag{4}$$

The parameter μ control shrinkage toward a multiple of the identity matrix. tr[.] is the trace operation. Friedman [4] proposed this as a principled way of dealing with singularity issue.

The influence of the negative examples can be tuned down by a discounting factor γ, and the discounted S_y is:

$$S_y^d = (1-\gamma)S_y + \frac{\gamma}{n} tr[S_y]I \tag{5}$$

With different combinations of the (μ, γ) values, regularized and/or discounted BDA provides a rich set of alternatives: ($\mu = 0$, $\gamma = 1$) gives a subspace that is mainly defined by minimizing the scatters among the positive examples, resembling whitening transform; ($\mu = 1$, $\gamma = 0$) gives a subspace that mainly separates the negative from the positive centroid, with minimal effort on clustering the positive examples; ($\mu = 0$, $\gamma = 0$) is the full BDA and ($\mu = 1$, $\gamma = 1$) represents the extreme of discounting all configurations of the training examples and doing nothing (with $W = I$) or anything (with arbitrary W).

3.4 Biased Discriminating Transform (BDT)

The solution for Equation (1) is the solution to a generalized eigenanalysis problem with the eigenvector matrix V associated with the (non-zero) eigenvalue matrix Λ satisfying the following equation:

$$S_y V = S_x V \Lambda \tag{6}$$

However, instead of using the eigenvector corresponding to the largest eigenvalue as in FDA (which has only one non-zero eigenvalue), more eigenvectors (*up to the* number of *effective dimensions*) can be retained with proper weighting to form the *discriminating transform matrix*

$$A = V\Lambda^{1/2} \tag{7}$$

The weighting of different eigenvectors by the square roots of their corresponding eigenvalues has nice properties as discussed in the next section, which makes BDT "a generalization of whitening transform with consideration of negative examples", or, a "*discriminative whitening transform*".

3.5 Properties of Discriminating Transform

Lemma 1 (Scatter Ratio Invariance) *Under any invertible square transformation W, the scatter ratio of Equation (1) does not change.* (*cf.* [5] p. 446)

This is because the determinants of W's canceled out for the square-shaped matrices.

Lemma 2 (Eigenvalue Invariance) *Under any invertible square transformation W on the data points, the generalized eigenvalues of the scatter matrix pair (S_y, S_x) will not change*.

The proof follows after expressing the scatter matrices in the transformed space as ($W^T S_y W$, $W^T S_x W$), and noticing that their eigenvalue matrix Λ_{new} satisfies

$$S_y(W\,V_{new}) = S_x(W\,V_{new})\Lambda_{new} \tag{8}$$

which indicates that Λ_{new} is the eigenvalue matrix of (S_y, S_x), and the eigenvectors are in the columns of ($W\,V_{new}$).

Theorem 1 (Fixed point in solution space up to a scale change) *After the first discriminating transform in the feature space, performing a second discriminating transform in this new space will yield a diagonal eigenvector matrix, i.e., identity matrix after normalization. The eigenvalues will remain the same.*

From Lemma 2 we know that $\Lambda_{new} = \Lambda$. Equation (8) also indicates that with proper scaling of the columns of V_{new}, we have $W\,V_{new} = V$, where V is the normalized eigenvector matrix of the first discriminnant transform. With $W = V\Lambda^{1/2}$, the un-normalized eigenvector matrix $V_{new} = \Lambda^{-1/2}$. After eigenvector normalization (so that each column has norm 1) it becomes the identity matrix.

This indicates that the first discriminating transform is the most critical one, which selects the projection directions in the descending order of discriminating power and weights them accordingly; and all subsequent transforms, if were to be performed, would only have the effect of further axis weighting in the new space. This leads to the following properties of the proposed transform:

Property 1 Step-by-step realization of dimension reduction *If performed iteratively, discriminating transform assigns relatively higher energy to more discriminative projection directions after each step.*

Figure 1(a) is a layout of positive and negative examples. (b) through (e) are the iterative biased discriminating transform (BDT) results, with (e) close to a direct "biased dimension reduction" onto the most discriminative direction, which is, in this case, the vertical axis of (e).

With non-representative negative examples, the learning machine shall not generalize too far into unlabeled area like a direct dimension reduction will do. A discriminating transform provides a way to moderate this process and to *trade off generalization for robustness*—although with a

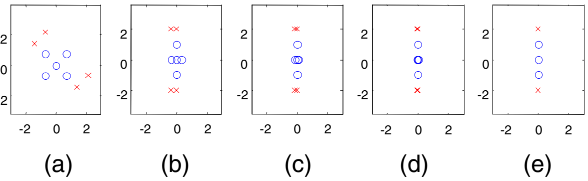


Figure 1 Illustration of Biased discriminating transform. Circles: positive class; crosses: negative class.

higher VC dimension [12] than a hyperplane, the quadratic boundary induced by the compactness constraint on positive examples insures robust performance *each round*.

Property 2 Whitening transform is a special case of BDT

Using the square roots of the eigenvalues to weight the corresponding eigenvectors is not arbitrary (even though it does not affect the value of the criterion by Lemma 1): BDT reduces to the whitening transform on positive examples when the contribution from negative examples is negligible or when the discounting factor γ is intentionally set to 1 (Equation (5) and (6)).

BDT selects not only the discriminative subspace, but also an optimal transform in this subspace to facilitate subsequent *nearest neighbor retrieval* that will maximize positive returns while minimizing the chance of confusing negative samples as positive ones.

3.6 Biased discriminant analysis using kernel (KBDA, or BiasMap)

Above analysis assumes Gaussian distribution on positive examples. To eliminate this assumption and to perform non-linear discrimination for non-linear data distributions, we adopt a kernel-based approach.

3.6.1 The kernel approach

The original linear BDA algorithm is applied in a "*feature space*" [1], $\mathcal{F}$, which is related to the original space by a non-linear mapping

$$\begin{aligned} \phi : \mathcal{C} &\rightarrow \mathcal{F} \\ x &\rightarrow \phi(x) \end{aligned} \tag{9}$$

where $\mathcal{C}$ is a compact subset of $\boldsymbol{R}^n$, such that originally linearly non-separable configurations becomes linearly separable in $\mathcal{F}$. However this mapping can be formidably expensive due to the arbitrarily large or even infinite dimension of $\mathcal{F}$, thus will not be carried out explicitly but through the evaluation of a kernel matrix K with components $k(x_i, x_j) = \phi^T(x_i)\,\phi(x_j)$. (*cf.* [12][1][7].)

3.6.2 BDA in kernel form—KBDA, or BiasMap

Using superscript ϕ to denote quantities in the new *feature space*[1], we rewrite the objective function as:

$$W_{opt} = \arg\max_{w} \frac{|W^T S_y^\phi W|}{|W^T S_x^\phi W|} \tag{10}$$

where

$$S_y^\phi = \sum_{i=1}^{N_y} (\phi(y_i) - m_x^\phi)(\phi(y_i) - m_x^\phi)^T \tag{11}$$

$$S_x^\phi = \sum_{i=1}^{N_x} (\phi(x_i) - m_x^\phi)(\phi(x_i) - m_x^\phi)^T \tag{12}$$

Here m_x^ϕ is the positive centroid in the *feature space*[1] $\mathcal{F}$. Since the solution w, column(s) of W, is the eigenvector(s) corresponding to the non-zero eigenvalues of the scatter matrices formed by the input vectors $\phi(x_i)$ and $\phi(y_j)$ in $\mathcal{F}$, the optimal w is in the subspace spanned by the input vectors. Thus w can be expressed as a linear combination of $\phi(x_i)$ and $\phi(y_j)$; and the problem of finding the optimal w becomes finding the optimal α with:

$$w = \sum_{i=1}^{N_x} \alpha_i \phi(x_i) + \sum_{j=1}^{N_y} \alpha_{j+N_x} \phi(y_j) = \Phi\alpha \tag{13}$$

where

$$\Phi = [\phi(x_1), \ldots, \phi(x_{N_x}), \phi(y_1), \ldots, \phi(y_{N_y})] \tag{14}$$

The numerator of (10), after expressed as the negative scatter with respect to positive centroid in $\mathcal{F}$, can be rewritten as:

$$\begin{aligned} w^T S_y^\phi w &= \alpha^T \Phi^T \sum_{j=1}^{N_y} (\phi(y_j) - m_x^\phi)(\phi(y_j) - m_x^\phi)^T \Phi\alpha \\ &= \alpha^T \sum_{j=1}^{N_y} (K_{y_j} - K_{mx})(K_{y_j} - K_{mx})^T \alpha \end{aligned} \tag{15}$$

Where

$$K_{y_j} = \Phi^T \phi(y_j), \qquad K_{mx} = \Phi^T m_x^\phi \tag{16}$$

Note that both of these are vectors of dimension $(N_x+N_y)\times 1$, and are in the dot-product forms suitable for the kernel evaluation. The summation term in the middle of Equation (15) can be further rewritten into

$$\begin{aligned} &\sum_{j=1}^{N_y} (K_{y_j} - K_{mx})(K_{y_j} - K_{mx})^T \\ &= (K_y - K_x I_{N_x}^y)(K_y - K_x I_{N_x}^y)^T \end{aligned} \tag{17}$$

where

$$K_y = \lfloor K_{y_1}, \ldots, K_{y_{N_y}} \rfloor, \qquad K_x = \lfloor K_{x_1}, \ldots, K_{x_{N_x}} \rfloor \tag{18}$$

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

Similarly, we can rewrite the denominator of (10),

$$\begin{aligned} w^T S_x^\phi w &= a^T (K_x - K_x I_{N_x}^x)(K_x - K_x I_{N_x}^x)^T a \\ &= a^T K_x (I - I_{N_x}^x) K_x^T \alpha \end{aligned} \tag{19}$$

here I is the identity matrix and $I_{N_x}^x$ is an $N_x \times N_x$ matrix of all elements being $1/N_x$.

3.6.3 The non-linear solution

To this point, by substituting Equations (15), (17), and (19) into (10) through (12), we arrive at a new generalized Rayleigh quotient, and it is again a generalized eigen-analysis problem, where the optimal □'s are the generalized eigenvectors associated with the largest eigenvalues λ's, i.e.,

$$\begin{aligned} &(K_y - K_x I_{N_x}^y)(K_y - K_x I_{N_x}^y)^T \alpha \\ &= \lambda K_x (I - I_{N_x}^x) K_x^T \alpha \end{aligned} \tag{20}$$

[1] A term used in kernel machine literatures to denote the new space after the nonlinear transform—this is not to be confused with the *feature space* concept previously used to denote the space for descriptors or features extracted from the media data.

With optimal □, the projection of a new pattern *z* onto *w,* ignoring weighting by square rooted eigenvalue, is directly given by:

$$w^T\phi(z)=\sum_{i=1}^{N_x}\alpha_i k(x_i,z)+\sum_{j=1}^{N_y}\alpha_{j+N_x}k(y_j,z) \qquad (21)$$

This projection is highly non-linear, i.e., points that are far apart (in Euclidean distance) in the original space can be arbitrarily close in the projected space [1][7].

3.6.4 The BiasMap algorithm

When applied in a retrieval system to facilitate relevance feedback, the algorithm is as follows:

Initially, the system retrieves initial results through whichever means available—query by keywords, sketch, or example, etc. In the subsequent round(s) with a set of positive and negative examples feedback by the user, and a chosen kernel *k*:

1. Compute K_x and K_y as defined in Equation (18). Degree of relevance or irrelevance can be incorporated much the same way as that in [8];
2. Solve for □ 's and λ's as in Equation (20);
3. With □ 's ordered according to the descending order of their eigenvalues, select the subset: { $\square_i$| $\lambda_i > \tau\lambda_1$}, where τ is a small positive number, e.g., 0.01;
4. $\square_i = \square_i \sqrt{\lambda_i}$, for selected $\square_i$ in the previous step;
5. Compute the projection of each point *z* onto the new space as in Equation (21).
6. In the new space, return the points corresponding to the Euclidean nearest neighbors from the positive centroid. Wait for user feedback then go to step 1.

Computation is light because it is non-iterative and the number of training samples is usually small.

4. Experiments and evaluations

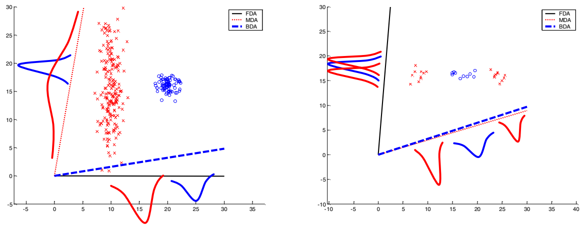
In this section we evaluate step-by-step the merit of the proposed scheme with the possible alternatives and state-of-the-art techniques, on both synthetic data and real world image databases.

4.1 Linear/Quadratic Case

For the linear versions of FDA, MDA, and BDA, all the transform matrices are linear, and the decision boundaries are either linear or quadratic.

4.1.1 Toy Problems

Toy problems are constructed to illustrate the first projection direction given by FDA, MDA, and BDA (Figure 2). Original data are in 2-D feature space, and positive examples are "o"s and negative examples are "x"s. FDA, MDA, and BDA are applied to find the best projection direction by their own criteria for each case, and the resulting eigenvector corresponding to the maximum eigenvalue is drawn in solid, dotted, and thicker dashed lines, respectively.



(a) (b)

Figure 2. Comparing FDA (solid), MDA (dotted), and BDA (thicker dashed) for dimensionality reduction from 2-D to 1-D. (a) FDA and BDA yield projections for better class separation, MDA (vertical) fails due to its effort to discriminate among negative examples; (b) MDA and BDA yield projection for class separation, FDA (vertical) fails completely due to its effort to cluster the two modes of negative examples.

In both cases, BDA yields good separation of negative examples from positive ones, as well as clustering of positive examples (it strikes a balance between these two goals, without any effort of modeling the density of negative examples). FDA and MDA are inadequate in biased classification or biased dimensionality reduction problem because of their forceful assumption on the number of modes for negative examples.

4.1.2 Image Database Testing

A COREL image set of 17695 images is tested. The feature space of 37 dimensions consists of 9 color moments [10], 10 wavelet moments [9], and 18 edge-based structure features [13]. Up to 20 rounds (or until convergence) of feedback are performed for each query under each of the four relevance feedback schemes: two-level optimal whitening transform (with partial independence assumption) on positive examples [8]—TWT, FDA, MDA, and BDA. Altogether over 1000 rounds of subject guided retrieval/relevance feedback are performed over 20 classes of images. The negative examples are selected by a subject during the retrieval. The numbers of hits in top 20 are recorded for different schemes. Their means and variances are listed in Table 1.

It is apparent that BDA not only yields the highest average score, but also has the minimum variation, which indicates the most robust performance. FDA and MDA have larger performance variation because they are affected by the clustering patterns in negative examples, which are generally unstable. MDA in this case is close to BDA in performance because the subject for this test tends to give small number (average around 3) of negative examples that are usually *not* from the same class. TWT

Table 1 Comparing relevance feedback results: the first row is the averaged number of hits in top 20, and the second row shows their variances.

No feedback	TWT	FDA	MDA	BDA
8.2	13.0	13.9	16.2	17.0
8.43	17.3	16.50	10.26	8.86

has low average score and large performance variation mainly because it is prone to be trapped at local minimum, which is frequently observed in our experiments. Using BDT the system can climb out of local minimum with the “push” from negative examples, and arrive at a better discriminating subspace. For example, when learning the concept of ‘elephant’ by positive examples alone, we see *grayish skin* and *long noses*; but seeing rhinos or hippos as the negative examples, we learn that *skin color* is much less important than *nose length*.

4.2 Non-linear Case

For the non-linear case, we first test whether the introduction of kernel helps or not; we then compare KBDA, or BiasMap, with kernel fisher discriminant analysis (KDA) [7] and SVM, over the same RBF kernel with varying spread parameters.

4.2.1 Does Kernel Help?

To test the ability of the KBDA in dealing with non-linear data configurations, synthetic experiments in two-dimensional space are used. In Figure 3 three schemes are compared: FDA, BDA, and KBDA. A significant boost in hit rates is observed when using KBDA.

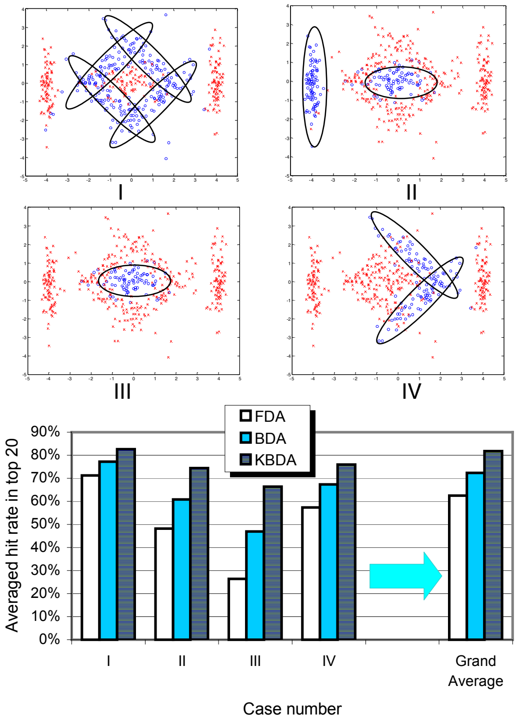


Figure 3 Test results on synthetic non-linear data configurations, which is constructed by a mixture of Gaussian modes—the ellipses roughly depict positive densities. The circles are positive examples and the crosses negative. A simulated query process is used for training sample selection, i.e., the 20 nearest neighbors of a randomly selected positive point are used as training samples. The bar diagram shows the averaged hit rate in top 20 returns for the four cases and a grand average.

Next we try automated testing of these algorithms on a fully labeled set of 500 images from COREL, with the aforementioned features. It consists of five classes, each with 100 images. Each round 10 positive and 10 negative images are randomly drawn as training samples. For each round the hit rate in the top 100 returns is recorded as the performance measures. 500 rounds of testing are performed on the 5 classes and the averaged hit rates are shown in Figure 4. KBDA or BiasMap outperforms others on average by a significant margin.

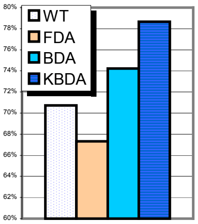


Figure 4 Averaged hit rates in top 100

4.2.2 KBDA, KDA and SVM

It is certainly desirable to see how the different kernel methods compare under different conditions.

The “Spillover” effect of KDA and SVM. First we use synthetic data to compare the three kernel machines, namely KBDA, KDA [7], and SVM, using the same RBF kernel and the same parameter set. The purpose is to see how they perform under different values of the spread parameter σ. Figure 5 shows the decision map for the given examples. It indicates that KBDA confines the positive region around the positive examples even with increasing σ of the RBF kernel, while KDA and SVM will “spillover” freely into part of the unlabeled areas of the feature space. For KDA, this is partly due to its effective transform dimension of only 1 (see Section 3.2); For SVM, this is in part due to the false assumption that the training examples are representative of the true distributions, which does not hold especially for negative examples (see argument in Section 3.1).

The spillover is dangerous since in the information retrieval application, given the small number of examples the unlabeled areas in the feature space are more likely to

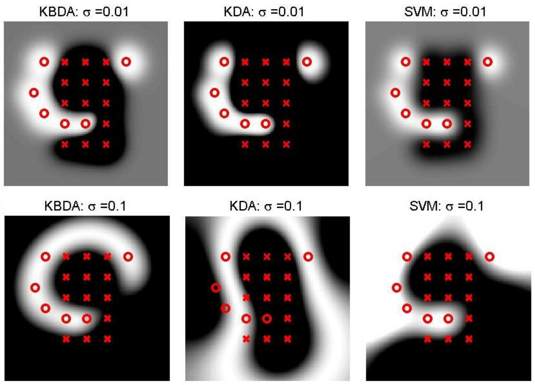


Figure 5 The decision surfaces of KBDA, KDA, and SVM for highly non-linear configurations. The open circles are positive examples and crosses negative. The gray level indicates the closeness to the positive centroid in the non-linearly transformed space: the brighter, the closer.

be negative. The spillover effect is most severe for SVM. When used for learning during information retrieval, the result of this effect is that after the user's feedback, the machine returns a totally different set of points, with most of them likely to be negative. This effect is further observed in the following experiment.

Small sample face and non-face classification: Finally KBDA, KDA, and SVM are compared in the context of face and non-face classification under a small number of training examples. A total of 1000 faces and 1000 non-face images are used. All the images are 16-by-16 in size and the original pixel values are used as the features, resulting in a 256-dimensional space. We use different numbers of positive and negative examples to train KBDA, KDA, and SVM learners. For SVM, two retrieval schemes are tested: *larger margin first* and *smallest positive margin first*. The percentage of face images in the top 1000 retrieved by SVM is used to compare with the precision of KBDA and KDA in their top 1000 returns.

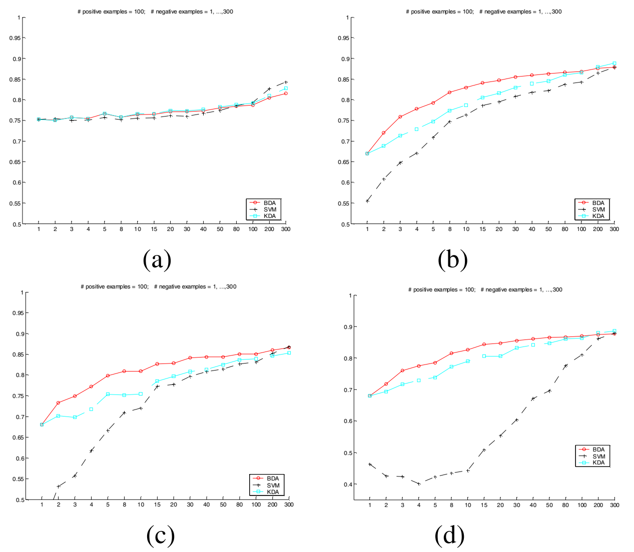
In Figure 6 each point on the curve represents the averaged precision of 100 trials, i.e., 100 runs of the algorithm over independently drawn random samples. In Figure 6(a), the spread parameter of the RBF kernel σ is set to be 1, which apparently is too small and the learning machines over-fit, i.e., even with more negative examples, the performance does not change significantly; and all three learning machines are similar in performance. This is the case illustrated by the top row in Figure 5.

In Figure 6(b) and (c), σ is set to be 50 and 100, respectively. When only one negative example is selected, KBDA reduces to KDA. When the number of negative examples is more than 1 but less than 100, we see that KBDA is more robust than KDA and SVM for changing values of σ; and that KBDA outperforms KDA and SVM. More experiments under other settings have led to the same conclusion, omitted here due to space limitations.

5. Summary

In this paper, we have taken a close look at the small sample learning problem during interactive multimedia retrieval. The key observation is the non-representative nature of the few training examples and the need for asymmetric treatment of the positive and the negative.

BiasMap is the nonlinear, kernel-based version of a novel variant of discriminant analysis, namely, *biased discriminant analysis*. It strikes a critical balance between informative and discriminative learning based on a limited number of training samples: compactness constrain (informative density modeling) is applied only on positive examples, while only discriminative constraint is imposed on negative examples. Considering the lack of information from limited training, such one-sided compactness assumption can be the best trade-off between robustness and generalization capability. Rigorous analysis along this direction is among our future research efforts.



(a) (b)

(c) (d)

Figure 6 Precision in top 1000 returns. Number of positive examples = 100, and the horizontal axis shows the changing number of negative examples from 1 up to 300. (a): σ = 1; (b) and (d): σ = 50; (c): σ = 100;

In (a)~(c) SVM returns the points with *larger margins first*; for (d) SVM returns the points with the *smallest but positive margins first*.

Acknowledgement: This work was supported in part by NSF Grant CDA 96-24396.

References

[1] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, 12(10): 2385--2404, 2000

[2] Y. Chen, X. S. Zhou, T. S. Huang, "One-class SVM for Learning in Image Retrieval", in *Proc. IEEE ICIP,* Greece, Oct, 2001

[3] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., New York, 1973

[4] J. Friedman, "Regularized Discriminant Analysis," *J. of Amer. Stat. Asso.*, vol 84, no. 405, pp. 165-175, 1989

[5] K. Fukunaga. *Introduction to statistical pattern recognition* (2nd ed). Academic Press, Boston 1990.

[6] Y. Ishikawa, R. Subramanya, and C. Faloutsos, "MindReader: Query databases through multiple examples", in *Proc. Of the 24th VLDB Conf.* New York, 1998

[7] S. Mika, G. Ratsch, and K.-R. Muller. "A mathematical programming approach to the Kernel Fisher algorithm," in *Advances in Neural Information Processing Systems* 13, 2001

[8] Y. Rui, T. S. Huang, "Optimizing learning in image retrieval", in *Proc. IEEE Conf. Comp. Vis. and Pat. Rec.*, Hilton Head Island, South Carolina, June, 2000, pp. 236-243

[9] J. R. Smith and S. F. Chang, "Transform features for texture classification and discrimination in large image databases," in *Proc. IEEE Int'l Conf. Image Proc.*, Austin, Oct. 1994

[10] M. Stricker and M. Orengo. "Similarity of color images," in *Proc. SPIE Stor. and Retrieval for Im. & Video Db.,* San Jose, 1995.

[11] Kinh Tieu and Paul Viola, "Boosting Image Retrieval", in *Proc. IEEE Conf. Comp. Vis. and Pat. Rec.*, SC, June, 2000.

[12] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995

[13] X. S. Zhou, T. S. Huang, "Edge-based structural feature for content-based image retrieval", *Pattern Recognition Letters*, Vol 22/5, Apr. 2001. pp 457-468.

[14] X. S. Zhou, T. S. Hunag, "Exploring the Nature and Variants of Relevance Feedback," in *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, Hawaii, Dec. 2001