Feature Selection in Example-Based Image Retrieval Systems

V. Shiv Naga Prasad
Dept. of Computer Science & Engineering,
Indian Institute of Technology - Madras,
Chennai - 600036.
shivnaga@cs.iitm.ernet.in

A.G. Faheema, Subrata Rakshit
Center for Artificial Intelligence and Robotics,
Raj Bhavan Circle, High Grounds,
Bangalore 560001, India.
(subrata,faheema)@cair.res.in

Abstract

The objective of Content Based Image Retrieval (CBIR) systems is to retrieve images from large datasets based on queries regarding their contents. This paper discusses the problem of selecting features for handling generic queries in Example-Based Image Retrieval, where the queries are given in the form of positive and negative examples. No assumptions are made regarding the nature of images or queries. We investigate several linear time-complexity algorithms which can be used for selecting features optimal for a given query. We test three aspects of the algorithms: their discrimination ability, robustness to sample set sizes and behavior in real world test cases. The results indicate that a hybrid approach may be called for.

1. Introduction

Example based query has become very popular in CBIR systems because of its intuitive appeal and ease of use. In this paradigm, the user query consists of examples of images he/she wants. The search engine then compares the images in the database with these to see whether they are apt or not. An extension to this method is to have positive as well as negative examples to aid in the search. A basic implementation of such a CBIR algorithm has been reported earlier in [3]. The present work is an extension of the feature selection module reported there.

In order to keep the system versatile, no assumptions are made regarding the nature of images and the types of queries that may need to be addressed. This requires the initial characterization of the images to be generic and redundant. A very large number of low level features are extracted from each image and kept in the database. The features are based on color, texture (both scale and orientation), gradient field, histograms etc. No decorrelation of the features is carried out. The number of features was 1076 in the first implementation and has increased even more as shape, orientation and other features were added. The feature selection module deals with the problem of choosing the subset of features appropriate for a given query. This information is then exploited by the search module to improve both speed and performance.

In this work we investigate several putative algorithms which can be used for feature selection. We discuss the feature selection problem in more detail before describing the various algorithms evaluated. Section 3 describes the selection methods evaluated. In Section 4 an evaluation procedure is outlined that assesses the selection methods using artificial data. Finally, performance for real queries on a real data set is reported and the results are analyzed.

2. Feature Selection Problem

The features selection module deals with the problem of choosing appropriate features for a given query, where the query is specified by positive and negative examples. Several approaches have been proposed to perform feature selection (see [1] for a brief survey). The most prominent of these are: hand-crafted systems [2] (these are not suitable for general example based queries); hierarchical systems [6], [4], where a-priori processing is needed to structure the database; self-organizing approaches [8], where the system continuously adapts itself to examples; Gaussian Mixture Model (GMM) based systems [5], where the class distributions are approximated using GMMs and then the discriminative power of the features is determined using Kullback-Leibler Divergence.

In the CBIR system in [3], where the feature selection module is to be used, the requirements from the module are as follows:

1. It must select a subset of features that provides the best input for the image selection module. If too many features are selected, the presence of irrelevant features will obscure the 'signal' (reduce the signal to noise ratio (SNR) [7]). On the other hand, taking too few features might impair the discrimination.
2. It must reduce the overall computational complexity by reducing the dimensionality of the classification problem.
3. Since we want the feature selection to take place at *every* query (i.e. within the user loop), it must be efficient. As the number of features is in the thousands, we require the mod-

ule to have linear time complexity with respect to number of features.
4. As we are dealing with generic high level queries, the module should not expect any a-priori organization of the images in the database to increase its efficiency.
5. The image selection module in [3] is also constrained to have linear time complexity w.r.t. number of features, therefore the feature selector should assume only linear discriminance based classifiers.
6. The module should be able to handle example sets of sizes as small as 5.

As can be seen, [2], [6], [4] do not meet requirement 4, [8] is not designed for the large number of features involved, and [5] cannot handle the small example set sizes which are available. Our present work addresses the feature selection problem in the context of the above mentioned requirements.

One important fact to note at this point is the difference in character between the positive and negative exemplar sets. The positive set (n_1 samples) necessarily represents a cohesive set with respect to some attribute. If the feature set is comprehensive enough, this attribute will be captured by one or more features. With respect to these features, the n_1 samples will most likely (but not necessarily) have a unimodal distribution. On the other hand, the negative examples (n_2 samples) are unlikely to belong to any well defined class. After a few iterations, this set consists of various images mistakenly retrieved by the CBIR system. Thus the n_2 samples may not have a unimodal distribution for the features of interest. The classifier design issues arising from the presence of two sets (and the effect of their non-Gaussian distributions) are discussed in [7].

3. Description of Algorithms Evaluated

In this section we describe the algorithms evaluated. Here onwards positive exemplars are referred to as 'n_1 samples' and negative examples are referred to as 'n_2 samples'

The first step is to generate several algorithms which will broadly cover the spectrum of possibilities. The desired properties of a feature selection method are: (a) linear time complexity with respect to number of features, i.e. it should score each feature independently and then take the best set of a predefined size; (b) low computational cost with respect to number of images in database; (c) ability to handle features with various types of distributions; and (d) robustness to low sample set sizes (down to even 5). Seven different methods were formulated and they can be divided into two broad categories: Statistical, and Entropy based.

The statistical methods evaluated were: delta-mean, inverse-sigma and membership-criterion. The Entropy based methods evaluated were: entropy of n_1 samples, entropy of both n_1 and n_2 samples, Kullback-Leibler distance (asymmetric), and Kullback-Leibler distance (symmetric).

We now describe each of these algorithms in detail.
In all the formulae in this section $\mu_{i,1}$ and $\mu_{i,2}$ are means of i^{th} features of n_1 and n_2 samples respectively; $\sigma_{i,1}$ and $\sigma_{i,2}$ are variances of i^{th} features of n_1 and n_2 samples respectively; $x_{i,j,1}$ and $x_{i,j,2}$ are the i^{th} features of j^{th} feature-vectors belonging to n_1 and n_2 samples respectively; and s_i is the score of the i^{th} feature.

3.1. Statistical Methods

These are methods which evaluate the relevance of features based on estimation of statistical parameters like mean and standard deviation. While these quantities are well defined for any distribution, their utility in characterizing distributions and estimating relevance of features depends on assumptions about the nature of the distributions. This section examines several such methods in detail.

Delta-Mean This is the only method used in the original implementation [3] and forms the starting point of our analysis. It measures the difference of means between the n_1 and n_2 samples, normalized by the sum of their standard deviations.

$$s_i = \frac{|\mu_{i,1} - \mu_{i,2}|}{\sigma_{i,1} + \sigma_{i,2}} \tag{1}$$

If the n_1 and n_2 clusters are well separated then the corresponding score is high, else it is low. The advantage of this method is that it is guaranteed to select only good features.The disadvantages are:(a) this scheme assumes that both the distributions are unimodal, (b) it is sensitive to errors in sigma(this occurs for very small n_1's and n_2's), and (c) it may miss good features, if either distribution is non-unimodal.

Since the n_2 distribution is most likely to violate the assumption of unimodality, the next logical step is to focus only on the n_1 distribution.

Inverse-Sigma Let us assume that only the n_1 distribution is unimodal. One way of characterizing features is to see how sharply the n_1 distribution is peaked. Since the comparison is between distributions (assumed to be unimodal) of identical sample set sizes, this can be measured by estimating the inverse of the standard deviation.

$$s_i = \frac{1}{\sigma_{i,1}} \tag{2}$$

Here onwards this criterion is called *InvSigma*. The advantage of *InvSigma* is that it does not assume n_2 distribution to be clustered. *InvSigma* will fail if n_1 distribution is also not unimodal. Of greater concern is the inability of *InvSigma* to utilize the information in the n_2 set, which is a key user feedback channel. The next method addresses this concern.

Membership Criterion This method restricts the assumption of unimodality to the n_1 set only and uses the

mean and standard deviation computed on the n_1 samples. However, these parameters are used in conjunction with the n_2 samples to determine a fitness score. The idea behind *Membership* is to see if the n_1 set parameters can be used to effectively differentiate n_1 samples from n_2 samples. Membership to n_1 set is defined as having a value within a certain distance ($\theta.\sigma_{i,1}$) from the n_1 set mean, $\mu_{i,1}$. The *Membership* score then measures the fraction of n_1+n_2 samples that are correctly assigned/not assigned to the n_1 set.

The actual implementation was as follows:

$$\text{let} \quad y_{i,j,1} = \frac{|x_{i,j,1} - \mu_{i,1}|}{\sigma_{i,1}} \quad \text{and} \quad y_{i,j,2} = \frac{|x_{i,j,2} - \mu_{i,1}|}{\sigma_{i,1}}$$

$\forall j$ do :

	$\leq \theta$	$> \theta$
$y_{i,j,1}$	increment a_{11}	increment a_{12}
$y_{i,j,2}$	increment a_{21}	increment a_{22}

$$s_i = \frac{a_{11} + a_{22}}{N} \tag{3}$$

where N = total number of vectors = $a_{11} + a_{12} + a_{21} + a_{22}$

We vary θ and get the best score possible for each feature. In our implementation θ is varied over {1, 1.5, 2, 2.5, 3, 3.5}

The advantage of *Membership* is that it does not make any assumptions about the n_2 sample distribution and yet takes its information into account. The disadvantage is that the n_1 sample distribution is required to be unimodal. Prima facie, this may seem like a non-issue for two reasons:(a) the assumption that n_1 sample set is unimodal may seem reasonable and (b) the constraint of linear discriminance may appear to rule out exploitation of non-unimodal distributions. However, neither of these are true. The user picks n_1 set as per some *high level* criteria which may not directly map to any low level features. (The requirement of generality rules out extraction of specific high level features.) Thus there is a significant probability of getting non-random multi-modal distributions for the n_1 set. Moreover, the image selection measure used in [3] is based on distances to *individual* exemplars. Thus that module is well constructed to exploit multi-modal distributions along individual features.

3.2. Entropy based methods

Statistical characterization of distributions always depends on assumptions about the distributions in order to be meaningful. When such assumptions cannot be made, an alternate approach is required. One such approach is an information theoretic approach, wherein the deviation from pure randomness is estimated by entropy of a distribution.

Sample entropy of n_1 In *Entropy(n_1)* the entropy of the (probability) distribution of each feature for the n_1 sample set is estimated. The intuition is that for a feature to be relevant the n_1 samples should be clustered, *i.e.,* the entropy of the distribution should be small. As the features are normalized to lie in [0,1] , we divide this interval into N equal sub-intervals. Let $p_{i,j}$ = probability of the i^{th} feature of n_1 samples to occur in the j^{th} sub-interval.

$$s_i = \log N + \sum_{j=1}^{N} (p_{i,j} \log p_{i,j}) \tag{4}$$

The advantage apparent in *Entropy(n_1)* is that it does not require the n_1 distribution to be unimodal or have any other parametric characterization. The disadvantage is that it only considers n_1 distribution and ignores the n_2 distribution.

Sample entropies of n_1 **and** n_2 In *Entropy(n_1, n_2)*, we use both the n_1 and n_2 distribution entropies for the score. The trivial extension of the previous method is to estimate the entropy of the n_2 distribution as well. This means that a good feature is one that gives a non-random distribution for n_1 as well as n_2 samples.

As before, the [0,1] interval is divided into N equal sub-intervals. Let $p_{i,j}$ = probability of the i^{th} feature of n_1 samples to occur in the j^{th} sub-interval. Let $q_{i,j}$ = probability of the i^{th} feature of n_2 samples to occur in the j^{th} sub-interval.

$$\begin{aligned} s_i &= 2 \log N + \sum_{j=1}^{N} (p_{i,j} \log p_{i,j}) \\ &\quad + \sum_{j=1}^{N} (q_{i,j} \log q_{i,j}) \end{aligned} \tag{5}$$

The disadvantage of *Entropy(n_1, n_2)* is that it loses sight of the ultimate objective, namely, discrimination of n_1 samples from n_2 samples. The score can be high even if the n_1 and n_2 distributions are non-random in *identical* ways.

KL distance There exists an entropy based measure of dissimilarity between two distributions, the Kullback-Leibler distance (divergence [1]). One may use this measure to determine how much the n_1 distribution differs from the n_2 distribution. Using same definitions as before,

$$s_i = \sum_{j=1}^{N} (p_{i,j} \log \frac{p_{i,j}}{q_{i,j}}) \tag{6}$$

Note that the above measure, *KL-Asymm*, is not symmetric in n_1 and n_2. This may be justified on the grounds that n_1 defines a coherent 'foreground' while the n_2 is a delimiter or 'background'. However, the measure can be made symmetric with minor modifications. The *KL-Symm* measure's score is computed as:

$$s_i = \sum_{j=1}^{N} (p_{i,j} \log \frac{p_{i,j}}{q_{i,j}}) + \sum_{j=1}^{N} (q_{i,j} \log \frac{q_{i,j}}{p_{i,j}}) \tag{7}$$

[1] It is not a metric in the strict sense, but in the present context we shall treat it as such.

4. Description of the Evaluation Procedure

At first glance it would seem that the KL-divergence metrics will easily outperform the other methods. However, they have two disadvantages: (1) high computational cost (we need to generate frequency histograms); and (2) they are not robust to small sample set sizes. Therefore we chose to evaluate all the methods uniformly without prejudice.

When testing the methods we found the following disadvantages in using real data: (1) it is very difficult to say which features are best for a given example set (i.e. ground truth is absent); (2) using real data, it is difficult to exhaustively try out all distribution possibilities; and (3) we will have small sample sets and it would be difficult to decouple the inherent limitations of a method from its sensitivity to sample set sizes.

These considerations prompted the formulation of a test procedure independent of the CBIR system that would use artificial data and focus exclusively on feature selection.

4.1. Generation of sample distributions

The n_1 and n_2 feature distributions are artificially generated, but in such a way as to model real distributions. Multiple n_1 and n_2 sample distributions are used which broadly cover all the possibilities. The generated distributions are of 6 main types:
d1: n_1 and n_2 samples form two distinct clusters.
d2: n_1 and n_2 samples form two overlapping clusters.
d3: n_1 samples form a tight cluster and n_2 samples are random (uniform distribution).
d4: Both n_1 and n_2 samples are random (uniformly distributed).
d5: n_2 sample distribution is bimodal and the two modes are on either side of a single (unimodal) n_1 sample cluster.
d6: n_1 samples are random (uniformly distributed) but n_2 samples form a cluster.
The reasoning behind **d1..d6** is the following: **d1** is the simplest and best distribution for discrimination; **d2** complicates it by having substantial overlap; **d3** and **d6** check whether a method can handle one of n_1 and n_2 samples being completely random; **d4** is the worst possible feature; and **d5** is the simplest case of multi-modal genre (i.e. one set is bimodal and the other unimodal). The clusters are modelled as Gaussian distributions. The variances of all the clusters are set to 0.15 and the samples are confined to the range [0,1]. In this section, the notation used is that a feature having a distribution **di** is labelled f(i).

4.2. Figures of merit for selection methods

We will now describe the figures of merit used to assess the algorithms.

Testing Ranking Ability: Features belonging to the different distribution types are given and each algorithm generates a ranking of the features. The following criteria are used to evaluate the ranking generated by each algorithm:
(A) Do f(1)'s have top scores? (Method can handle simplest and most favorable case)
(B) Do f(5)'s have top scores? (Method can pick good non-unimodal features)
(C) Are f(4)'s at the bottom ? (Method can reject irrelevant features)
(D) Do f(2)'s,f(3)'s and f(6)'s score higher than f(4)'s ? (Method can distinguish non-ideal features from irrelevant features.)
(E) Are f(2)'s graded worse than the f(1)'s ? (Method considers n_2 samples, and hence overlaps)
Methods are credited +1 for positive answers, -1 for negative answers and 0 if results are inconclusive.

Testing Sensitivity to Sample Set Sizes: We also investigated the sensitivity of the methods to sample set sizes. This was done because the estimation of means, variances and entropies becomes erroneous as the number of samples decreases. The sensitivity of methods to set sizes varies according to the type of parameters used and the nature of use. To evaluate this aspect we checked how well scores for **d1** distributions were separated from the corresponding scores for **d4** distributions, for different set sizes. The intuition being that **d1**'s are the ideal features and **d4**'s, the worst possible. The means and variances of the distributions were kept constant. The more the separation for low sample set sizes the better it is since it would mean that the algorithm is less sensitive to low set sizes. To assess this, we defined a figure of merit, $\Delta\tilde{\mu}_i$, for every method (i).

$$\Delta\tilde{\mu}_i = \frac{\tilde{\mu}_{i,1} - \tilde{\mu}_{i,4}}{\tilde{\sigma}_{i,1} + \tilde{\sigma}_{i,4}}$$

Where $\tilde{\mu}_{i,1}$ and $\tilde{\mu}_{i,4}$ are the means, and $\tilde{\sigma}_{i,1}$ and $\tilde{\sigma}_{i,4}$ are the standard deviations of the scores generated by method(i), of features belonging to **d1** and **d4** distribution types respectively.

Testing Performance for Real Queries and Data: Last but not the least, an attempt was made to evaluate the effect of various feature selection methods on the full CBIR system. The test was done with one user in the loop and assessed how well various methods are able to exploit user feedback. For various queries, the number of iterations required for retrieving 50 images was determined. This was done for different sizes of positive and negative example sets (taken from the same database).

5. Experimental Results

For the first two experiments involving simulations with artificial data, the scoring is done only once (i.e. no iterations)

Rank	Delta-Mean	InvSigma	Membership	Entropy(n_1)	Entropy(n_1,n_2)	KL-Symm	KL-Asymm
1^{st}	f(1) [1.00]	f(1) [1.00]	f(5) [1.00]	f(1) [1.00]	f(1) [1.00]	f(1) [1.00]	f(1) [1.00]
2^{nd}	f(1) [.999]	f(1) [.994]	f(5) [.999]	f(2) [.992]	f(1) [.995]	f(1) [.998]	f(1) [.997]
3^{rd}	f(2) [.559]	f(2) [.989]	f(1) [.984]	f(1) [.991]	f(2) [.981]	f(5) [.897]	f(5) [.991]
4^{th}	f(2) [.556]	f(5) [.974]	f(1) [.982]	f(5) [.982]	f(2) [.977]	f(5) [.896]	f(5) [.985]
5^{th}	f(3) [.003]	f(5) [.973]	f(2) [.974]	f(5) [.977]	f(5) [.856]	f(2) [.406]	f(2) [.398]
6^{th}	f(5) [.002]	f(2) [.970]	f(2) [.971]	f(3) [.975]	f(5) [.851]	f(2) [.396]	f(2) [.396]
7^{th}	f(5) [.0013]	f(3) [.964]	f(3) [.943]	f(2) [.971]	f(6) [.495]	f(6) [.256]	f(6) [.372]
8^{th}	f(4) [.0007]	f(3) [.963]	f(3) [.936]	f(3) [.969]	f(3) [.491]	f(3) [.255]	f(6) [.364]
9^{th}	f(3) [.0006]	f(6) [.364]	f(4) [.759]	f(4) [.008]	f(6) [.489]	f(3) [.253]	f(3) [.141]
10^{th}	f(6) [.0005]	f(4) [.364]	f(4) [.755]	f(6) [.008]	f(3) [.489]	f(6) [.252]	f(3) [.138]
11^{th}	f(4) [.0004]	f(6) [.362]	f(6) [.448]	f(6) [.008]	f(4) [.007]	f(4) [.002]	f(4) [.002]
12^{th}	f(6) [.0001]	f(4) [.361]	f(6) [.446]	f(4) [.007]	f(4) [.007]	f(4) [.001]	f(4) [.001]

Table 1: Scores and ranks generated by the seven algorithms under study.

and no user interaction is involved. The first experiment is to see how the algorithms compare with each other, given a fixed number of samples. In order to eliminate any influence of sample set size on the ranking ability the number of samples was fixed at 10,000. Twelve feature distributions are generated, consisting of two per category (**d1..d6**). The twelve scores computed by each method are normalized so that the best score is 1.00. The detailed results are given in Table 1. The twelve distributions are separately ranked by each method, as shown in each column. Each entry indicates the type of distribution and its normalized score.

Based on the results shown in Table 1, the seven methods are evaluated as per the five criteria listed earlier in Sec 4.2. The results are as shown in Table 2.

	A	B	C	D	E	Total
Delta-Mean	+1	-1	+1	-1	+1	1
Inverse Sigma	+1	+1	+1	-1	-1	1
Membership	+1	+1	0	-1	+1	2
Entropy(n_1)	+1	+1	0	-1	-1	0
Entropy(n_1,n_2)	+1	0	+1	+1	-1	2
KL-Symm	+1	+1	+1	+1	+1	5
KL-Asymm	+1	+1	+1	+1	+1	5

Table 2: The points earned by each method as per the results shown in Table 1. The criteria are listed in Sec 4.2.

It can be seen that the KL divergence methods score highest. The reason is that these methods do not make any assumptions about unimodality or any particular kind of distribution. Additionally, these methods fully utilize the information available in both n_1 and n_2 sample sets. Amongst the statistical methods, *InvSigma* performs as well as the original *Delta-Mean* method while the *Membership* method outperforms both. It is important to note that in these artificial tests there is no user interaction.

The second experiment is performed to investigate robustness with respect to small n_1 and n_2 sample set sizes. Recall that the first experiment had 10000 samples in each set so as to negate any effect due to sample set sizes. In practice, sample set sizes can start from as small as 5 and grow with each iteration. For simplicity, only three of the seven methods are tested in the second experiment. These three are: *(i) KL-Symm*, which is the best; *(ii) Membership*, which is the best statistical method and *(iii) InvSigma*, which depends on the least number of parameter estimates. The behavior of the three selected methods was observed for sample set sizes of $\{10, 50, 100, 500, 1000, 10000\}$.

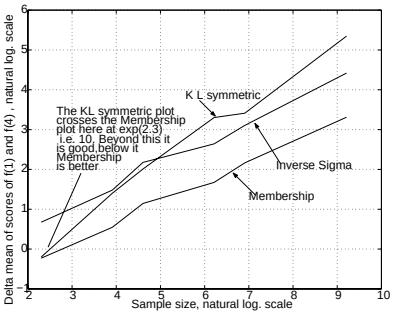

Figure 1: Graph of $\Delta\tilde{\mu}_i$ versus the sample set sizes. Reduction in set sizes affects the ability of all methods to discriminate between good and bad features (f(1) and f(4)). This deterioration is not same for all methods. The relative merits of the methods *cannot* be judged from this graph alone.

It can be seen from Figure 1 that the *KL-Symm* algorithm gives separable clusters for sample set sizes greater than 33. The reason for such behavior is that in order to calculate entropies of distributions we need to quantize the data and generate frequency distributions. This operation becomes highly inaccurate as the number of samples becomes small. The *InvSigma* method is most robust with respect to sample set size reduction. As the set sizes reduce, *KL-Symm* first becomes worse than *InvSigma* and finally becomes worse

than *Membership*. We can see from the Figure 1 that for samples set sizes greater than 10 the *KL-Symm* algorithm fares better than *Membership* algorithm. Clearly, a hybrid strategy may be called for. The system may start off using *InvSigma* or *Membership* (when n_1 and n_2 sets are small) and switch over to *KL-Symm* as the set sizes build up. The actual choice of methods depends on the merits of a method as well as its suitability for a given sample set size.

The third experiment investigates the merits of various selection methods in the context of actual CBIR application. The database chosen is a set of 10,000 images of varying sizes and types (outdoors, artificial, portraits, textures, signs etc). Three different queries are tried. *(i)* Garden flowers: Images with red flowers against a backdrop of greenery. *(ii)* Sea: Images having land/water combinations or open sea. *(iii)* Sports cars: Images of racing cars. Each query is started with initial n_1 and n_2 sets of three sizes - 5/5, 10/10, 20/20. The average number of iterations required to retrieve 50 acceptable images is shown in Table 3. In some cases only the n_2 set increased dramatically and the search had to be stopped as a failure. These are marked as X.

	5,5	10,10	20,20
Delta-Mean	7	6	5
InvSigma	X	X	X
Membership	7	5	4
KL-Symm	X	5	3

Table 3: Number of iterations required for extracting 50 images for various methods. Columns represent various initial (n_1, n_2) set sizes . Results are average of 3 queries.

When used in a real application, the inability of the *InvSigma* method to utilize the n_2 sample information proves fatal. The *KL-Symm* method does very well for large sample set sizes but breaks down completely at the smallest size. The *Membership* method fares better than the *Delta-Mean* method and deteriorates gracefully as sample set size decreases. Although *KL-Symm* seems to produce results comparable to those by *Membership* at size of 10 itself, the computational cost of this algorithm may not warrant its use for sizes less than 20. It is recommended that the *Membership* method be used initially and the *KL-Symm* method be invoked later as sample set sizes approach 20.

6. Conclusions

The choice of a feature selection method plays a critical part in the success of a versatile CBIR system. Such systems owe their versatility to a highly redundant feature set. In order to optimize the system in real time to a specific query, it is necessary to have a fast feature selection module that can adapt the system to each query at each iteration.

The present work investigates seven different methods for scoring the features. All of them have linear computational complexity w.r.t. number of features. Its results highlight three desirable attributes of any feature selection method: (1) The method should not make any assumption regarding the distribution of a feature value in either the positive or negative exemplar sets; (2) The method should utilize the information in *both* sets; and (3) The method should not require large exemplar sets.

The seven methods presented in this work are first calibrated as per desirable properties as feature selectors for discrimination tasks. Three of the seven methods are then examined for robustness to sample set size variation. Finally, these three, along with the original *Delta-Mean* method, are tested in a CBIR system with real data and high level queries. The results indicate that an optimal solution may be a hybrid one where the *Membership* method is used initially for sample set sizes less than 20 and the *KL-Symm method* is invoked later for larger sets.

7 Acknowledgements

This work was sponsored by CAIR, Bangalore. The authors wish to thank Director, CAIR, for his support. Shiv Naga worked at CAIR under the aegis of the Summer Student Project Training Program (May-June 2001).

References

[1] A. Jain and D. Zongkar. Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. and Machine Intell.*, 19(2):153–158, 1997.

[2] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, New Jersey, 1988.

[3] C. V. Jawahar, P. J. Narayanan, and S. Rakshit. A flexible scheme for representation, matching, and retrieval of images. In *Proc. of ICVGIP 2000*, pages 271–277. Allied Publishers Ltd., 2000.

[4] R. Ng and D. Tam. Multi-level filtering for high-dimensional image data: Why and how. *IEEE Trans. Knowledge and Data Engg.*, 11(6), Dec. 1999.

[5] J. Novovikova, P. Pudil, and J. Kittler. Divergence based feature selection for multimodal class densities. *IEEE Trans. Pattern Anal. and Machine Intell.*, 18(2):218–223, Feb. 1996.

[6] D. L. Swets and J. J. Weng. Hierarchical discriminant analysis for image retrieval. *IEEE Trans. Pattern Anal. and Machine Intell.*, 21(5):386–401, 1999.

[7] Q. Tian, Y. Wu, and T. S. Huang. Incorporating discriminant analysis with EM algorithm in image retrieval. In *Proc. of IEEE 2000 Int. Conf. on Multimedia and Expo (ICME2000), Hilton NY Towers, NY*, volume 1, pages 299–302, Jul 2000.

[8] J. Wang, N. Ahuja, and T. S. Huang. Learning recognition and segmentation using the cresceptron. In *Proc. Int. Conf. on Computer Vision, Berlin, Germany*, pages 121–128, May 1993.