Feature Relevance Estimation for Image Databases

Jing Peng*and Bir Bhanu
Center for Research in Intelligent Systems
University of California, Riverside, CA 92521
Email: {jp,bhanu}@vislab.ucr.edu

Abstract

Content-based image retrieval methods based on the Euclidean metric expect the feature space to be isotropic. They suffer from unequal differential relevance of features in computing the similarity between images in the input feature space. We propose a learning method that attempts to overcome this limitation by capturing local differential relevance of features based on user feedback. This feedback, in the form of accept or reject examples generated in response to a query image, is used to locally estimate the strength of features along each dimension. This results in local neighborhoods that are constricted along feature dimensions that are most relevant, while enlongated along less relevant ones. We provide experimental results that demonstrate the efficacy of our technique using real-world data.

1 Introduction

The rapid advance in digital imaging technology makes possible the wide spread use of image libraries and databases. This in turn demands effective means for access to such databases. It has been well documented that simple textual annotations for images are often ambiguous and inadequate for image database search. Thus, retrieval based on image "content" becomes very attractive [1, 3, 5, 6]. Generally, a set of features (color, shape, texture, etc.) are extracted from an image to represent its content. As such, image database retrieval becomes a K nearest neighbor (K-NN) search in a multidimensional space defined by these features under a given similarity metric.

Simple K nearest neighbor search, as an image retrieval procedure, returns the K images closest to the query. Obviously this involves the issue of measuring the closeness or similarity between two images. The most common measure of the similarity between two images is the distance between them. If the Euclidean distance

$$D(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{q}(x_i - y_i)^2} \tag{1}$$

is used, then the K closest images to the query $\mathbf{x}_Q$ are computed according to $\{\mathbf{x}|D(\mathbf{x},\mathbf{x}_Q) \leq d_K\}$, where d_K is the Kth order statistic of $\{D(\mathbf{x}_i,\mathbf{x}_Q)\}_1^N$. Here N is the number of images in the database. The major appeal for the simple K-NN search method resides in its ability to produce continuous and overlapping rather than fixed neighborhoods, and to use a different neighborhood for each individual query so that all points in the neighborhood are close to the query.

One problem with the Euclidean metric, however, is that it does not take into account the influence of the scale of each feature variable in the distance computation. Changing the scale of a feature dimension in different amounts alters the overall contribution of that dimension to the distance computation, hence its influence in the nearest neighbors retrieved. This is usually considered undesirable. An additional limitation is that the use of the Euclidean distance, while simple computationally, implies that the input space is isotropic. However, the assumption for isotropy is often invalid and generally undesirable in many practical applications.

In this paper, we propose a novel method that provides a solution to the problems discussed above. With this method an image retrieval system is able to learn differential feature relevance in an efficient manner by estimating the strength of each feature dimension in predicting a given query. In addition, since the estimation process is carried out locally in the vicinity of the input query, the method is highly adaptive to query locations.

*Current address: Computer Science Department, Oklahoma State University, Stillwater, OK 74078. Email: jpeng@cs.okstate.edu

2 System Overview

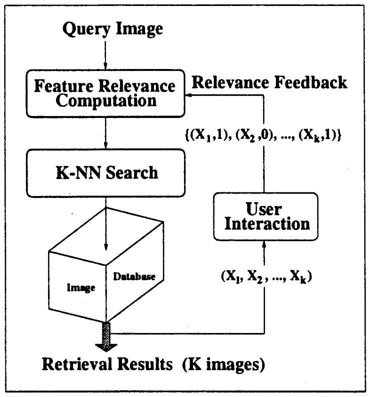

Figure 1: System for learning feature relevance.

Figure 1 shows the functional architecture of our system. Images in the database are represented by feature vectors, such as normalized mean and standard deviations of responses from Gabor filters. The user presents a query image to the system. At this time feature relevance along each dimension is assumed to be equal, and its associated weighting is initialized to $1/q$, where q is the dimension of the feature space. The system carries out image retrieval using a K-NN search based on current weightings to compute the similarity between the query and all images in the database, and returns the K nearest images. The user then marks the retrieved images as positive (class 1) (e.g., click on an image by using the left mouse button) or negative (class 0) (e.g., click on an image by using the right button). The user "thinks" that the positive images look similar to the query image but the negative ones do not. Note that in practice, only images that are dissimilar to the query (negative images) need to be marked. These marked images constitute training data. From the query and the training data, the system computes local feature relevance in terms of weights, from which a new round of retrieval begins. The above process repeats until the user is satisfied with the results or the system cannot improve the results from one iteration to the next.

3 Weighted K-Nearest Neighbor Search

Simple K-NN search clearly has its limitations as a procedure for content-based image retrieval. The objective of our approach is to develop a retrieval method that inherits the appealing properties of K-NN search, while at the same time overcomes its limitations by capturing the notion of local feature relevance.

3.1 Local Feature Relevance

The retrieval performance for image databases can be characterized by two key factors. *First*, for a given query image, the relevance of all the features input to the database system may not be equal for retrieving similar images. Irrelevant features often hurt retrieval performance. *Second*, feature relevance depends on the location at which the query is made in the feature space. Capturing such relevance information is a prerequisite for constructing successful retrieval procedures in image databases.

We note at the outset that this problem is opposite to typical classification problems based on lazy learning techniques, such as nearest-neighbor kernel methods. While the goal in classification is to predict the class label of an input query from nearby samples, the goal in retrieval is to find samples having the same "class label" as that of the query. Moreover, many lazy learning techniques for classification lend themselves to the kind of problems retrieval tasks may face. It is important to realize that, unlike classification problems, the notion of classes in image databases is a user-centered concept that is dependent on the query image. There is no labelling of images in the database. Nonetheless, the "class label" of an image is simply used here as a vehicle to facilitate the theoretical derivation of our feature relevance measure for content-based image retrieval. As we shall see later, the resulting relevance measure and its associated weightings are independent of image labels in the database and, thus, fit our goals nicely here.

3.2 Local Relevance Measure

We begin this section by introducing some classification concepts essential to our probabilistic derivation.

In a two class (1/0) classification problem, the class label $y \in \{0,1\}$ for query $\mathbf{x}$ is treated as a random variable from a distribution with the probabilities $\{\Pr(1|\mathbf{x}), \Pr(0|\mathbf{x})\}$ [2]. We then have

$$f(\mathbf{x}) \doteq \Pr(1|\mathbf{x}) = \Pr(y=1|\mathbf{x}) = E(y|\mathbf{x}), \tag{2}$$

To predict y at $\mathbf{x}$, $f(\mathbf{x})$ is first estimated from a set of training data using techniques based on regression, such as the least-squares estimate[1]. Decision tree methods, neural networks and nearest-neighbor kernel methods are examples of using this regression paradigm to the classification problem. The Bayes classifier can then be applied to achieve optimal classification performance. In image retrieval, however, the "class label" of $\mathbf{x}$ is known, which is 1 in terms of the notation given above. All that is required is to exploit the differential relevance of input features for image retrieval. Consider the least-squares estimate for $f(\mathbf{x})$. In the absence of values for any variable assignments, it is simply

$$E[f(\mathbf{x})] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}. \tag{3}$$

That is, the optimal prediction (in the least-squares sense) for $f(\mathbf{x})$ is its average value. Now given only that $\mathbf{x}$ is known at dimension $x_i = z$. The least-squares estimate becomes

$$E[f(\mathbf{x})|x_i = z] = \int f(\mathbf{x})p(\mathbf{x}|x_i = z)d\mathbf{x}. \tag{4}$$

Here $p(\mathbf{x}|x_i = z)$ is the conditional density of the other input variables defined as

$$p(\mathbf{x}|x_i = z) = p(\mathbf{x})\delta(x_i - z) / \int p(\mathbf{x})\delta(x_i - z)d\mathbf{x}, \tag{5}$$

where $\delta(x-z)$ is the Dirac delta function having the properties $\delta(x-z) = 0$ if $x \neq z$ and $\int_{-\infty}^{\infty} \delta(x-z)dx = 1$. It is evident that

$$0 \leq E[f|x_i = z] \leq 1.$$

Furthermore, Equation (4) shows the predictive strength (probability) once the value of just one of the input features x_i is known.

Let $\mathbf{z}$ be the query. Since $f(\mathbf{z}) = 1$ (recall that query $\mathbf{z}$ has the same class label (class one) as the positive images), it follows that

$$f(\mathbf{z}) - 0$$

is the largest error one makes in predicting f at $\mathbf{z}$. That is, $f(\mathbf{z}) - 0$ is the error incurred when one predicts $f(\mathbf{z})$ to be zero ($\mathbf{z}$ is not in class one), but in fact $f(\mathbf{z})$ is in class one with probability 1. On the other hand

$$f(\mathbf{z}) - E[f(\mathbf{x})|x_i = z]$$

is the error one makes by predicting the probability of $\mathbf{z}$ being in class one as $E[f(\mathbf{x})|x_i = z]$, conditioned on feature x_i taking the value z. Then

$$[(f(\mathbf{z}) - 0) - (f(\mathbf{z}) - E[f(\mathbf{x})|x_i = z])] = E[f(\mathbf{x})|x_i = z] \tag{6}$$

[1]Note that one can also use techniques based on density estimation to compute $f(\mathbf{x})$.

represents a reduction in error between the two predictions. We can now define a measure of feature relevance for query $\mathbf{z}$ as

$$r_i(\mathbf{z}) = E[f(\mathbf{x})|x_i = z]. \tag{7}$$

That is, feature x_i is more relevant for query $\mathbf{z}$ if it contributes more to the reduction in prediction error (6).

The relative relevance, as a weighting scheme, can then be given by

$$w_i(\mathbf{z}) = (r_i(\mathbf{z}))^t / \sum_{l=1}^{q} (r_l(\mathbf{z}))^t. \tag{8}$$

where $t = 1, 2$, giving rise to linear and quadratic weightings, respectively. In this paper we propose the following exponential weighting scheme

$$w_i(\mathbf{z}) = \exp(Tr_i(\mathbf{z})) / \sum_{l=1}^{q} \exp(Tr_l(\mathbf{z})) \tag{9}$$

where T is a parameter that can be chosen to maximize (minimize) the influence of r_i on w_i. When $T = 0$ we have $w_i = 1/q$, thereby ignoring any difference between the r_i's. On the other hand, when T is large a change in r_i will be exponentially reflected in w_i. In this case, w_i is said to follow the Boltzmann distribution. The exponential weighting is more sensitive to changes in local feature relevance (7) and gives rise to better performance improvement, as we shall see later.

It is clear from (8) and (9) that $0 \leq w_i(\mathbf{z}) \leq 1$, where $w_i(\mathbf{z}) = 0$ indicates that knowing x_i at $\mathbf{z}$ does not to help predict the query. On the other hand, $w_i(\mathbf{z}) = 1$ states that having the knowledge of x_i at $\mathbf{z}$ is sufficient to predict the query. Values in between show the degrees of relevance that x_i exerts at $\mathbf{z}$. It reflects the influence of feature x_i on the variation of $f(\mathbf{x})$ at query location $\mathbf{z}$. Thus, (8) and (9) can be used as weights associated with features for weighted similarity computation

$$D(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{q} w_i (x_i - y_i)^2}. \tag{10}$$

It can be shown that (10) is indeed a metric. These weights enable the similarity computation to elongate less important feature dimensions, and, at the same time, to constrict the most influential ones.

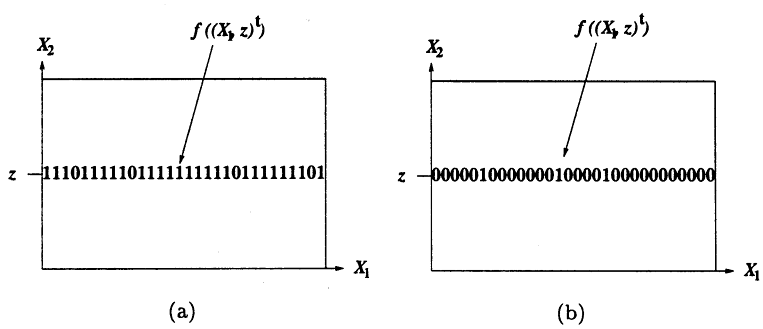

Figure 2: (a): Large $E[f(\mathbf{x})|x_2 = z]$ implies that the subspace spanned by X_1 at z is likely to contain samples having the same class label as the query. (b): Small $E[f(\mathbf{x})|x_2 = z]$ indicates that the subspace is unlikely to have samples similar to the query.

A justification for (7) and, hence, (8) and (9), goes like this. Suppose that the value of $E[f(\mathbf{x})|x_i = z]$ (4) is large, which implies a large weight along dimension x_i. This, in turn, penalizes points along x_i that are moving away from z. Now $E[f(\mathbf{x})|x_i = z]$ can be large only if the subspace spanned by the other input dimensions at $x_i = z$ likely contains samples coming from class 1, assuming a uniform distribution. This scenario is illustrated in Figure 2(a). Then a large weight assigned to x_i based on (8) says that moving away from the subspace, hence from the data in class 1, is a bad thing to do. Similarly, a small value of $E[f(\mathbf{x})|x_i = z]$, hence a small weight, indicates that in the vicinity of x_i at z one is unlikely to find samples similar to the query, as shown in Figure 2(b). Therefore, in this situation in order to find samples resembling the query, one must look farther away from z.

4 Estimation of Relevance

In order to estimate (8) and (9), one must first compute (7). The retrieved images with relevance feedback from the user can be used as training data to obtain estimates for (7), hence (8) and (9). Let $\{\mathbf{x}_j, y_j\}_1^K$ be the training data. Here $\mathbf{x}_j$ denotes the feature vector representing jth retrieved image, and y_j is either 1 (relevant) or 0 (irrelevant) marked by the user as the class label associated with $\mathbf{x}_j$. To compute $E[f(\mathbf{x})|x_i = z]$, recall that $f(\mathbf{x}) = E[y|\mathbf{x}]$. Thus, it follows that

$$E[f(\mathbf{x})|x_i = z] = E[y|x_i = z]$$

However, since there may not be any data at $x_i = z$, the data from the vicinity of x_i at z are used to estimate $E[y|x_i = z]$, a strategy suggested in [2]. Therefore, (7) can be estimated according to

$$\hat{E}[y|x_i = z] = \sum_{j=1}^{K} y_j 1(|x_{ji} - z| \leq \Omega) / \sum_{j=1}^{K} 1(|x_{ji} - z| \leq \Omega), \tag{11}$$

where $1(\cdot)$ is an indicator function. That is, $1(\cdot)$ returns 1 if its argument is true, and 0 otherwise. Ω can be chosen so that there are sufficient data for the estimation of (4). In this paper, Ω is chosen such that

$$\sum_{j=1}^{K} 1(|x_{ji} - z| \leq \Omega) = C, \tag{12}$$

where $C \leq K$ is a constant. It represents a trade-off between bias and variance. In addition, (11) can be computed within a subregion, thus making the relevance measure more local. Note that it is possible to extend this technique to multiple class situations where the user can grade the retrieved images. Our retrieval and feature relevance computation algorithm is summarized in Figure 3, where *precision* denotes the average retrieval precision.

1. Let i be current query; initialize weight vector $\mathbf{w}$ to $\{1/q\}_1^q$.
2. Compute K nearest images using $\mathbf{w}$.
3. User marks the K images as positive or negative.
4. While *precision* $< \theta$ Do
   (a) $Tset \leftarrow Tset \cup \{marked\ K\ images\}$.
   (b) Update $\mathbf{w}$ from Eqs. (11) and (9) using training data in $Tset$.
   (c) Compute K nearest images using $\mathbf{w}$.
   (d) User marks the K images as positive or negative.

Figure 3: The probabilistic feature relevance learning (PFRL) Algorithm.

The bulk of the computational expense involved in the feature relevance estimation algorithm shown in Figure 3 is consumed by the K-nearest neighbor search, while the relevance estimation is quite efficient.

This is particularly pronounced when the image database is large. In practice, however, the amount of computation associated with the nearest neighbor search can be significantly reduced by partitioning or indexing the image database in such a way that the nearest neighbor search can be localized within a given partition. This issue, however, is beyond the scope of this paper.

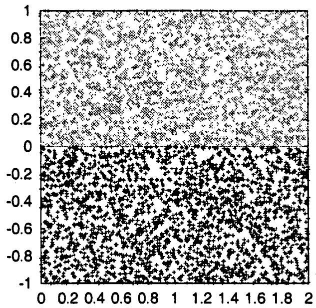

Figure 4: A simple two class problem with a uniform distribution. The red square indicates the query.

We use a simple two class problem, shown in Figure 4, to illustrate the feature relevance computation process. In this problem, the data for both classes are generated from a uniform distribution. The number of data points for both classes is roughly the same. The red square, located at $(1, 0.1)$, represents the query. Figure 5(a) shows the 200 nearest neighbors (red squares) of the query found by the unweighted K-NN method (1). The resulting shape of the neighborhood is circular, as expected. In contrast, Figure 5(b) shows the 200 nearest neighbors of the query, computed by the technique described above. That is, the retrievals (with relevance feedback) shown in Figure 5(a) are used to compute (7) and, hence, (9) with estimated new weights: $w_1 = 0.134$ and $w_2 = 0.866$. As a result, the new (elliptical) neighborhood is elongated along the horizontal axis (the less important one) and constricted along the vertical axis (the more important one). The effect is that there is a dramatic increase in the retrieved nearest neighbors that are similar to the query.

This example demonstrates that even a simple problem in which the class boundary ideally separates two classes can benefit from the feature relevance learning technique just described, especially when the query approaches the class boundary. It is important to note that for a given distance metric the shape of a neighborhood is fixed, independent of query locations. Furthermore, any distance calculation with equal contribution from each feature variable will always produce spherical neighborhoods. Only by capturing the relevant contribution of the feature variables can a desired neighborhood be realized that is highly customized to query locations.

5 Empirical Results

In the following we compare two competing retrieval methods using real data. (a): The probabilistic feature relevance learning (PFRL) algorithm described above; and (b): The MARS algorithm [6]. Note that there is a third method, the simple (unweighted) K-NN method, that is being compared against implicitly. The first retrieval by the two methods is based on the unweighted K-NN method. Also, in all the experiments, features representing data are first normalized to lie between 0 and 1, and the performance is measured using the average retrieval precision [5, 6].

5.1 The Problems

Database 1. The data set (SegData), taken from the UCI repository [4], consists of images that were drawn randomly from a database of 7 outdoor images. The images were hand segmented by the creators of the database to classify each pixel. Each image is a region. There are 7 classes, each of which has 330 instances. Thus, there are total 2310 images in the database. These images are represented by 19 real valued attributes.

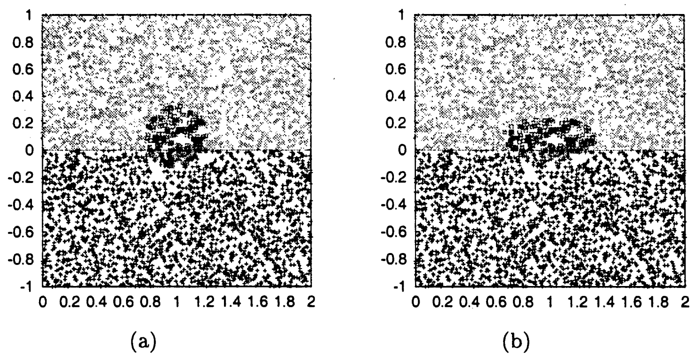

Figure 5: Effect of feature relevance on retrieval. (a) Circular neighborhood (no learning). (b) Elliptical neighborhood (after learning).

Database 2. The data are obtained from MIT Media Lab at: whitechapel.media.mit.edu/ pub/VisTex. There are total 640 images of 128×128 in the database with 15 classes. The images in this database are represented by 8 Gabor filters (2 scales and 4 orientations).

Database 3. Letter Image Recognition Data (LIRD) taken from [4] consists of a large number of black-and-white rectangular pixel arrays as one of the 26 upper-case letters in the English alphabet. The characters are based on 20 Roman alphabet fonts. They represent five different stroke styles and six different letter styles. Each letter is randomly distorted through a quadratic transformation to produce a set of 20,000 unique letter images that are then converted into 16 primitive numerical features.

Database 4. This is a set, also taken from [4], of 208 data points having two classes (Mines and Rocks) with equal number of instances in each class. The data are represented by 60 features. For details, see [4].

5.2 Results

For all the problems, each image in the database is selected as a query and top 20 (corresponding to parameter K in the algorithm (Fig. 3) described above) nearest neighbors are returned that provide necessary relevance feedback. Note that only negative images (that are different from the query determined by the user) need to be marked in practice. The average retrieval precision is summarized in Table 1. There are two rows under each problem in the table, comparing the performance of the two competing methods.

The second column in Table 1 shows the average retrieval precision obtained by the method without any relevance feedback (rf). That is, it is the results of applying the simple K-NN method using unweighted Euclidean metric. The third column and beyond show the average retrieval precision computed, at the specified iterations, after learning has taken place. That is, relevance feedback obtained from the previous retrieval is used to estimate local feature relevance, hence a new weighting. The procedural parameters input to the algorithms under comparison were determined empirically that achieved the best performance. They are by no means exhaustive. For the SegData image database, they were set to 13 and 19 (PFRL), and 0.1, 0.6 and 0.1 (MARS); for the MIT image database, they were set to 13 and 20 (PFRL), and 0.1, 0.6 and 0.1 (MARS); for the LIDR data, they were set to 14 and 19 (PFRL), and 0.1, 0.6 and 0.1 (MARS); for the Sonar data, they were 11 and 25 (PFRL), and 0.1, 15.0 and 0.8 (MARS), respectively.

It can be seen from Table 1 that both methods demonstrate significant performance improvement across the tasks. In general, PFRL seems to outperform MARS on all the tasks. Superior performance by PFRL is particularly pronounced on the SegData and Sonar data sets. Moreover, the results show that PFRL can handle large problems with high dimensionality well by its superb performance on the 60 dimensional Sonar data set.

Table 1: Average retrieval precision for real data.

SegData Data Set					
Method	0 (rf)	1 (rf)	2 (rf)	3(rf)	4 (rf)
PFRL	90.90	94.67	95.78	96.49	96.79
MARS	83.01	86.41	87.38	87.12	87.81
MIT Data Set					
Method	0 (rf)	1 (rf)	2 (rf)	3 (rf)	4 (rf)
PFRL	81.14	85.30	86.95	88.71	89.46
MARS	83.66	86.72	87.58	87.93	88.18
LIRD Data Set					
Method	0 (rf)	1 (rf)	2 (rf)	3 (rf)	4 (rf)
PFRL	76.85	84.59	87.34	89.13	89.97
MARS	77.61	84.34	86.26	86.95	87.25
Sonar Data Set					
Method	0 (rf)	1 (rf)	2 (rf)	3 (rf)	4 (rf)
PFRL	61.61	82.50	90.05	93.29	94.38
MARS	66.30	75.94	78.39	79.50	79.69

6 Conclusions

This paper presented a novel probabilistic feature relevance learning technique for efficient content-based image retrieval. The experimental results using large real data show convincingly that learning feature relevance based on user's feedback can indeed improve retrieval performance of an image database system. Furthermore, since the relevance estimate is local in nature, the resulting retrieval, in terms of the shape of the neighborhood, is highly adaptive and customized to the query location.

Acknowledgements

This work was supported by DARPA/AFOSR grant F49620-97-1-0184. The contents of the information do not necessarily reflect the position or the policy of the U.S. Government.

References

[1] M. Flickner et al., "Query by Image and Video Content: The QBIC system" *IEEE Computer*, pp. 23-31, September 1995.

[2] J.H. Friedman "Flexible Metric Nearest Neighbor Classification," Tech. Report, Dept. of Statistics, Stanford University, 1994.

[3] T.P. Minka and R.W. Picard, "Interactive Learning with a "Society of Models"", *Pattern Recognition*, vol.30, (no.4):565-81, April 1997.

[4] P.M. Murphy and D.W. Aha, UCI repository of machine learning databases. 1995. www.cs.uci.edu/~mlearn/MLRepository.html.

[5] J. Peng, B. Bhanu and S. Qing, "Probabilistic Feature Relevance Learning for Content-Based Image Retrieval," *Computer Vision and Image Understanding*, 1999.

[6] Y. Rui, T.S. Huang and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS," Proceedings of IEEE International Conference on Image Processing, pp. 815-818, Santa Barbara, California, October, 1997.