Learning Integrated Online Indexing for Image Databases*

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

Abstract

Most of the current image retrieval systems use "one-shot" queries to a database to retrieve similar images. Typically a K-NN (nearest neighbor) kind of algorithms is used where weights measuring feature importance along input dimensions remain fixed (or manually tweaked by the user) in the computation of a given similarity metric. However, the similarity does not vary with equal strength or in the same proportion in all directions in the feature space emanating from the query image. The manual adjustment of these weights is time consuming and exhausting. Moreover, it requires a very sophisticated user. In this paper we present a novel method that enables image retrieval procedures to continuously learn feature relevance based on user's feedback, and which is highly adaptive to query locations. Experimental results are presented that provide the objective evaluation of learning behavior of the method for image retrieval.

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 is well known that simple textual annotations for images are often ambiguous and inadequate for image database search. Thus, retrieval based on image "content" becomes very attractive [3, 5, 6]. Generally, a set of features (color, shape, texture, etc.) are extracted from an image to represent its content. Then the image database retrieval procedure becomes a K-NN search in a multidimensional space defined by the set of features under a given similarity metric, such as the Euclidean distance.

There are several fundamental problems associated with this simple content-based image retrieval scheme. *First*, features are unequal in their differential relevance for computing the similarity between images. Feature relevance may change from image to image, and from location to location [1]. When a user says that two images are similar, the user really means that the images are similar in an individual feature, some combination of the features, or some features still unknown to the user. This implies that the similarity does not vary with equal strength or in the same proportion in all directions in the feature space emanating from the query image. Figure 1 illustrates a case in point, where class boundaries are along coordinate axes. At query a dimension X is more relevant while at query b dimension Y is. However, at query c both dimensions are equally relevant.

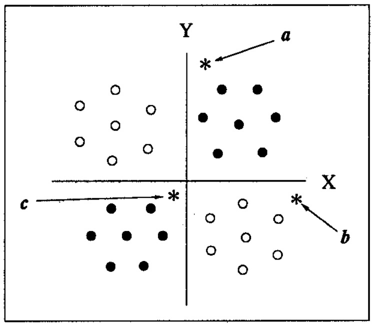


Figure 1: Feature relevance varies with query locations.

Second, the user understands more about the query, whereas the database system can only "guess" what the user is looking for during the retrieval process. The system must interact with the user to learn feature differential relevance to guide its search and to iteratively refine its retrieval at run-time.

Finally, different similarity (closeness) measures capture different aspects of perceptual similarity between images [5]. Humans do seem to selectively attend to features to optimize their visual behaviors [7]. In general, what similarity metric to use is image de-

*This work was supported by DARPA/AFOSR grant F49620-97-1-0184 at the University of California, Riverside.

0-8186-8821-1/98 $10.00 © 1998 IEEE

pendent, and plays an important role in the outcome of the retrieval process. For example, in the context of "eigenfaces" for recognition, the quadratic or *Mahalanbis* distance is more appropriate on the assumption of a Gaussian distribution. While determining similarity metrics is an important research issue, we are mainly concerned with learning feature relevance here.

In this paper, we present a novel method that enables image retrieval systems to learn differential relevance of features in an efficient manner, and which is highly adaptive to query images. Next we describe the functional architecture of our retrieval system. We then introduce the notion of local feature relevance and ways to estimate it. Finally, we show experimental results demonstrating the efficacy of our technique using both simulated and real-world data.

1.1 System Overview

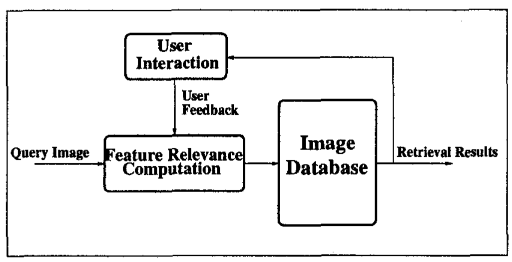


Figure 2: System for learning feature relevance.

Figure 2 shows the functional architecture of our system. Images in the database are represented by feature vectors whose components are normalized mean and standard deviations of responses from Gabor filters. The user presents a query image to the system. At this time feature relevance weighting is initialized to $1/q$, where q is the size of the feature vectors. The system then carries out image retrieval using unweighted K-NN search to compute similarity between the query and all images in the database, and returns top K nearest neighbors. The user then marks the retrieved images as positive (e.g, click on an image by using the left mouse button) or negative (e.g, click on an image by using the right button). The user "thinks" that positive images look similar to the query image but the negative ones don't. These marked images constitute training data. From the query and training data, the system computes local feature relevance (weights), from which a new round of retrieval begins. The above process repeats until the user is satisfied with the results.

2 Local Feature Relevance

The performance of retrieval 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 retrieval procedure 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 local relevance is essential for constructing successful retrieval procedures in image databases.

2.1 Local Relevance Measure

In a two class (1/0) classification problem, the class label y at query $\mathbf{x}$ is treated as a random variable from the distribution $\{\Pr(1|\mathbf{x}), \Pr(0|\mathbf{x})\}$. We then have [4]

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

To predict y at $\mathbf{x}$, $f(\mathbf{x})$ is estimated from a set of training data using regression techniques, such as least squared estimation. In image retrieval, however, the "label" of $\mathbf{x}$ is known, which is 1 in terms of the notation given above. All that is required is to exploit differential relevance of the input features to image retrieval. Consider the least-squares estimate for $f(\mathbf{x})$, given only that $\mathbf{x}$ is known at dimension $x_i = z$. It is simply

$$E[f|x_i = z] = \int f(\mathbf{x}) p(\mathbf{x}|x_i = z) d\mathbf{x}. \tag{2}$$

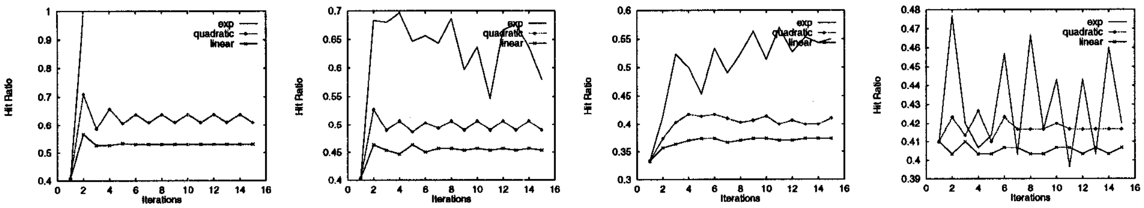
Here $p(\mathbf{x}|x_i = z)$ is the conditional density of the other input variables defined as

$$p(\mathbf{x}|x_i = z) = p(\mathbf{x}_z) / \int p(\mathbf{x}) \delta(x_i - z) d\mathbf{x} \tag{3}$$

where $\mathbf{x}_z = (x_1, \cdots, x_{i-1}, z, x_{i+1}, \cdots, x_q)^T$, and $\delta(x-z)$ is the Dirac Delta function having properties $\delta(x-z) = 0$ for $x \neq z$ and $\int_{-\infty}^{\infty} \delta(x-z)dx = 1$. Eq. (2) shows the prediction strength (probability) once the value of just one of the input features x_i is known. A feature relevance measure can then be given by

$$r_i(\mathbf{z}) = (E[f|x_i = z_i])^t / \sum_{l=1}^{q} (E[f|x_l = z_l])^t. \tag{4}$$

where $t = 1, 2$, giving rise to linear and quadratic weighting, respectively. It is also possible to exponentially weigh feature relevance: $r_i(\mathbf{z}) = \exp(E[f|x_i = z_i]) / \sum_{l=1}^{q} \exp(E[f|x_l = z_l])$. One can see that $0 \leq r_i(\mathbf{z}) \leq 1$. $r_i(\mathbf{z}) = 0$ when $f(\mathbf{x})$ does not depend on x_i at query $\mathbf{z}$. And $r_i(\mathbf{z}) = 1$ when $f(\mathbf{x})$ depends entirely on x_i at $\mathbf{z}$. Values in between show the degrees of relevance that x_i exerts at $\mathbf{z}$. Thus, (4) can be used as



(a) query 1 (b) query 2 (c) query 3 (d) query 4

Figure 3: Performance of exponential, quadratic, and linear weightings on simulated data.

weights associated with features for weighted similarity metric computation. These weights elongate less important dimensions, and at the same time constrict the most influential ones.

The intuition behind (2), and hence (4), is as follows. Suppose that the value of $E[f|x_i = z]$ (2) is large (close to 1), which means that in the subspace spanned by the other input dimensions at $x_i = z$ one is more likely to find samples coming from class 1, assuming a uniform distribution. Then a large weight assigned to x_i based on (4) says that moving away from the subspace, hence from data in class 1, is a bad thing to do. Similarly, a small value of $E[f|x_i = z]$ indicates that in the vicinity of x_i at z one is unlikely to find samples similar to the query. Therefore, in order to find samples resembling the query, one must look farther from z.

2.2 Estimation

In order to estimate (4), one must first compute (2). The retrieved images with feedback from the user can be used as training data to obtain estimates for (2) and (4). Let $\{\mathbf{x}_j, y_j\}_1^N$ be the training data, where $\mathbf{x}_j$ denotes the feature vector representing jth retrieved image, and y_j is either 1 or 0 marked by the user as a desired (positive or hit) or undesired (negative) image, respectively. Then, based on (1), one can estimate (2) [4] according to

$$\hat{E}[y|x_i = z] = \sum_{j=1}^{N} y_j 1(|x_{ji} - z| \leq \Omega) / \sum_{j=1}^{N} 1(|x_{ji} - z| \leq \Omega) \tag{5}$$

where $1(\cdot)$ is an indicator function. Ω can be chosen so that there are sufficient data for the estimation of (2). In addition, (5) can be computed within a subregion, thus making the relevance measure more local. Note that this technique can be readily extended to multiple class situations where the user can grade retrieved images.

3 Empirical Evaluation

In order to evaluate our algorithm we performed a number of experiments on both simulated and real world data. The simulated data experiment allows us to reliably predict the strengths and limitations of our algorithm because the precise nature of the problem the algorithm is facing is known. Note that for the experiments reported in this paper, the weighted Euclidean metric is used.

3.1 Experiments on Simulated Data

This problem is designed in such a way that all feature dimensions have the same global relevance. However, they have unequal local differential relevance depending on query locations. There are $q = 5$ feature dimensions and 2 classes. The data is generated from a normal distribution $\mathbf{x} \sim N(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ is given by $\mathbf{\Sigma} = diag\{0.75^2\}_1^q$. The classes are defined by $\sum_{i=1}^{5} x_i^2 \leq 2.3 \Rightarrow class\ \ 0, otherwise \Rightarrow class\ \ 1$. There are 100,000 data points in the database with roughly equal number of data points in each class.

Table 1: Four representative queries.

Query	Coordinates				
q_1	0.01	0.005	0.02	0.005	1.531
q_2	0.01	0.02	0.015	1.081	1.081
q_3	0.03	0.08	0.88	0.88	0.88
q_4	0.02	0.76	0.76	0.76	0.76

Queries from class 1 are generated such that some feature dimensions are more relevant than others. Four representative queries are shown in Table 1. Clearly, feature x_5 is the most discriminating dimension for query q_1. Similarly, features x_4 and x_5 are the most influential dimensions for query q_2, and features x_3, x_4 and x_5 for query q_3, and so on. Within a neighborhood of each representative query, additional four query points are randomly generated. We want to see how our algorithm will behave in each representative region. Top 300 nearest points are returned for each query. The number of points in Ω to estimate (5) is 100.

Figure 3 graphs the hit ratio (the number of positive retrievals divided by the total number of retrievals) as a function of time. Note that at the first iteration, the hit ratio is produced by unweighted K-NN. The results show that our method can indeed capture local feature relevance. In addition, it outperforms unweighted K-NN. It is interesting to note that the improvement is particularly pronounced when the number of relevant features is small. As the number of relevant features increases, our technique reduces to unweighted K-NN, as expected. Figure 4 illustrates weight change as the number of iterations increases.

3.2 Experiments on Real Data

There are 10,314 texture images of size 64×64 in our database that are represented by 48 dimensional feature vector. We use 24 Gabor filters (4 scales and 6 orientations) for feature extraction. The mean μ_{mn} and the standard deviation σ_{mn} of the magnitude of the transform coefficients are used as feature components of a feature vector after being normalized by the standard deviations of the respective features over the entire feature database.

As shown in Figure 2, the system receives feedback from user interaction, and the user may not be consistent during the interactive process. This requires a large scale subjective experiment with various users to obtain average results. However, in this paper we focus on objective (rather than subjective) evaluation of our system. In our system, the 10,314 images are partitioned into 220 classes. There are about 45 images in each class. These 220 classes form the *ground truth* for our experiments. For each query image, the system returns 40 nearest images based on its current weighted similarity computation. From the retrieved images, the system chooses those as positive examples that come from the same class as the query and uses rest of them as negative examples. In practice, however, these positive examples will be identified by the user. These marked samples constitute the training data. The termination conditions for the retrieval procedure are (i) the number of positive examples returned exceeds a given certain threshold (25 in these experiments); (ii) the number of iterations reaches 3. A sample query image and corresponding positive and negative example images are shown in Fig. 5.

Table 1 shows sample retrieval results by using different similarity metrics where learning feature relevance improves performance over time. The first row indicates class labels associated with each query. The second row shows the number of positive hits received after the first iteration. Note that this number is a result of simple, unweighted K-NN search. The third row shows the number of positive hits after the second iteration. Similarly, the results for other iterations are shown. The retrieval results show clear improvement over time due to learning.

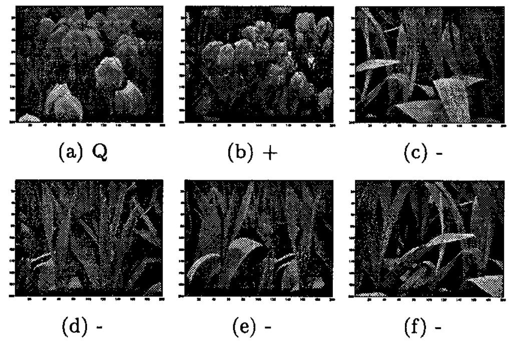


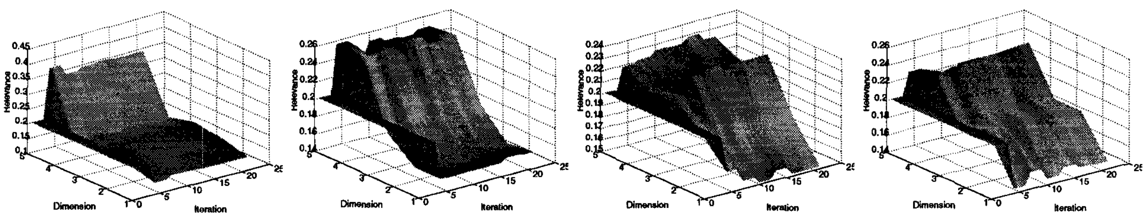
Figure 5: (a): Sample query (Q). (b): Sample positive example (+). (c)-(f): Sample negative examples (-).

4 Related Work

Friedman [4] describes an approach for learning local feature relevance that combines some of the best features of K-NN learning and recursive partitoning. This approach recursively homes in on a query along the most (locally) relevant dimension, where local relevance is computed from a reduction in prediction error given that query's value at that dimension. This method performs well on a number of classification tasks. In contrast, our method, inspired by [4], computes local feature relevance directly from the conditional probabilities, since in a retrieval problem the "label" of the query is known.

PCF [2], *per category feature importance*, is another example that computes feature relevance based on conditional probabilities. This method estimates the conditional probability $p(c|f)$ for feature f in every category and uses it as a weight for f in category c. PCF assigns large weights to features having high correlation with the class. Clearly, feature importance as such is non-local, and therefore, insensitive to query locations.

Trott and Leng [9] introduce an *integer programming* (IP) method for computing feature weights for an interactive retrieval task. The method imposes a



(a) (b) (c) (d)

Figure 4: Feature weight change as a function of iteration. (a) q_1, (b) q_2, (c) q_3, (d) q_4.

Table 2: Retrieval results.

Iters\Class	106	169	190	156	53	127	207	38	146	211	132	213	103	72	55	151	175	134	206	83	46	62	47	192	216	114	52
0	21	14	22	18	19	16	15	21	16	20	19	23	22	22	19	21	22	13	19	17	16	20	21	23	19	18	23
1	31	24	24	27	26	26	24	26	25	27	24	29	30	32	25	30	33	27	30	28	24	27	26	29	25	32	33
2		26	25				25				25										26						
3																											

fixed ratio between the weight of the last feature and the sum of the weights of all other features for training samples. While this method could be used for interactive retrieval, its performance is not yet known.

5 Conclusions

This paper shows that learning feature relevance based on user's feedback can indeed improve retrieval performance of an image database system. The knowledge acquired during one retrieval can be gradually collected and it can become part of the database itself through continuous learning. This knowledge can be used in conjunction with case-based learning [1, 8] to achieve generalizarion in future retrievals in order to further optimize the performance of the system.

References

[1] D. W. Aha, D. Kibler and M. K. Albert, "Instance-based Learning Algorithms," *Machine Learning*, 6:37-66, 1991.

[2] R.H. Creecy, B.M. Masand, S.J. Smith, and D.L. Waltz, "Trading Mips and Memory for Knowledge Engineering," *CACM,* 35:48-64, 1992.

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

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

[5] R. Jain, S.N.J. Murthy, and L. Tran, "Similarity Measures for Image Databases," *IEEE Conference on Fuzzy Logic*, 1995.

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

[7] R. M. Nosofsky, "Attention, Similarity, and the Identification-Categorization Relationship," *Journal of Experimental Psychology: General 15* :39-57, 1986.

[8] J. Peng, J., "Efficient Memory-Based Dynamic Programming," Proceedings of the 12th International Conference on Machine Learning, pp. 438-446, Tahoe City, California, July 1995.

[9] J.R. Trott and B. Leng, "An Engineering Approach for Troubleshooting Case Bases," Proceedings of the 2nd ICCBR conference, pp. 178-189, Providence, RI. Springer, 1997.

[10] X. Wu and B. Bhanu, "Gabor Wavelet Representation for 3-D Object Recognition," *IEEE Trans. on Image Processing*, 6(1), pp. 47-64, Jan. 1997.