FAST ADAPTIVE MAHALANOBIS DISTANCE-BASED SEARCH AND RETRIEVAL IN IMAGE DATABASES

Sharadh Ramaswamy and Kenneth Rose

Signal Compression Lab
Dept. of Electrical and Computer Engineering
University of California, Santa Barbara
CA 93106 - 9560
{rsharadh,rose}@ece.ucsb.edu

ABSTRACT

Motivated by the need to efficiently leverage user relevance feedback in content-based retrieval from image databases, we propose a fast, clustering-based indexing technique for exact nearest-neighbor search that adapts to the Mahalanobis distance with a varying weight matrix. We derive a basic property of point-to-hyperplane Mahalanobis distance, which enables efficient recalculation of such distances as the Mahalanobis weight matrix is varied. This property is exploited to recalculate bounds on query-cluster distances via projection on known separating hyperplanes (available from the underlying clustering procedure), to effectively eliminate noncompetitive clusters from the search and to retrieve clusters in increasing order of (the appropriate) distance from the query. We compare performance with an existing variant of VA-File indexing designed for relevance feedback, and observe considerable gains.

Index Terms— Relevance feedback, similarity search, image indexing, vector quantization, clustering

1. INTRODUCTION

Advancements in semiconductor technology, magnetic storage hardware and the growth of the Internet has spawned new database applications for multimedia data, such as Multimedia Information Systems, CAD/CAM, Geographical Information systems (GIS), medical imaging that store large amounts of data periodically in and later, retrieve it from databases. Searching over such image and multimedia databases is primarily performed under the content-based image search and retrieval (CBIR) paradigm. Images are typically represented by feature vectors and the measure of similarity between two images is assumed to be proportional to the distance between their feature vectors. Recently, a combination of texture features (extracted through Gabor filters) and color features (histograms) have been found to be efficient descriptors for a broad class of images and form a part of the MPEG-7 multimedia standard (see [1]).

Useful feature vectors are often high dimensional, such as the 60 dimensional texture descriptors of [1]. The search for nearest neighbors in large, high dimensional data sets is challenging. The search time is overwhelmingly dominated by IO operations (e.g., hard disk access times). Index structures exist that facilitate search and retrieval of multi-dimensional data, but it has been observed that the performance of many such structures degrades with increase in dimensionality. In a famous result, Weber et. al. [2] have shown that whenever the dimensionality is above 10, these methods are outperformed by a simple sequential scan. The reason for this degradation in performance is attributed to Bellman's celebrated '*curse of dimensionality*' [3], which refers to the exponential growth of hyper-volume with dimensionality of the space.

1.1. Relevance Feedback in Image Retreival

While the Euclidean distance metric is popular within the multimedia indexing community, it is by no means the perceptually "correct" distance measure. Hence, significant research activity (in content-based image retrieval) has been directed toward Mahalanobis (or weighted Euclidean) distances (see [4]). The Mahalanobis distance measure has more degrees of freedom than the Euclidean distance and by proper updation (or *relevance feedback*), has been found to be a much better estimator of user perceptions (see [5, 6, 4]).

The goal in relevance feedback is to adapt the distance measure to match user expectations, by making the search an interactive process. Here, in each iteration a set of results is retrieved and user provides feedback on the relevance of each result. If Mahalanobis distance is employed, this is used to update the weight matrix for the next iteration. Sometimes, the query vector is also modified [5]. The process stops when

This work was supported in part by NSF, under grants IIS-0329267 and CCF-0728986, the University of California MICRO program, Applied Signal Technology Inc., CISCO Systems Inc., Dolby Laboratories Inc., and Qualcomm Inc.

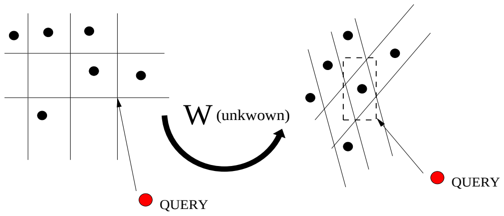

Fig. 1. VA-File under an Unknown Linear Transformation

the user is satisfied with the results.

1.2. Efficiency of Indexing with Relevance Feedback

Multidimensional search indexes are typically designed assuming a fixed Mahalanobis distance measure that is known in advance. The weight matrix is diagonalized and the data are correspondingly rotated and scaled into a new set of dimensions, prior to indexing. However, in relevance feedback applications, the weight matrix changes with time and renders most standard indexes ineffective and very slow. Clearly, a truly effective relevance feedback application requires a new indexing approach.

A very popular and effective technique employed to overcome the curse of dimensionality is the Vector Approximation File (VA-File) [2]. VA-File partitions the space into hyper-rectangular cells, aligned with the co-ordinate axes. Each dimension is quantized uniformly and the quantization indices are stored of each feature vector in the so called *approximation file*, on the hard-disk. Upper and lower bounds on the distance to the query from each cell are estimated and these are used to prune the data-set of those vectors that are not likely to be good candidates. The final set of candidate vectors are read from the hard-disk and the nearest neighbor are determined.

A change in the Mahalanobis weight matrix is equivalent to rotating and skewing the bounding rectangles into uniform hyper-parallelograms. The method of [7] fits minimum bounding rectangles that contain these parallelograms. Note that these hyper-rectangles are larger and overlapping. A new set of distance bounds to these rectangles are evaluated and used in spatial filtering (see Figure 1).

In this paper, we consider a clustering approach towards similarity search. The data set is clustered using a standard clustering or vector quantization (VQ) technique and only relevant ("nearest") clusters are retrieved during query processing. Clusters are retrieved until the k^{th} nearest neighbor discovered so far is closer to the query than all remaining clusters, which guarantees that the k nearest neighbors have been discovered. We further note that with only one cluster, the indexing technique degenerates to the sequential scan i.e. sequential scan is a *special case*. Central to such a search technique is the ability to tightly bound the distance to a cluster, without accessing the elements of the cluster [8]. We show how effective estimates of query-cluster distances can be performed while adapting to a *changing* weight matrix. and how this filters out irrelevant regions of the database, thus providing significant speed-ups over known techniques. Consequently, the proposed clustering based approach is effective for relevance feedback in image databases.

2. POINT-TO-HYPERPLANE DISTANCE

Let $d_W(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T W(\mathbf{x}-\mathbf{y})}$ be the distance between any two feature vectors $\mathbf{x}$ and $\mathbf{y}$. Without loss of generality, we assume W is symmetric and positive definite i.e. $d_W(\cdot,\cdot)$ is a metric. Let $H(\mathbf{a},b) = \{\mathbf{x} : \mathbf{a}^T\mathbf{x} + b = 0\}$ be a hyperplane and $\mathbf{y}$ a point in the space outside of it. Then,

$$d_W(\mathbf{y},H) = \min_{\mathbf{x}\in H} d_W(\mathbf{x},\mathbf{y}) = \sqrt{\min_{\mathbf{x}\in H} d_W(\mathbf{x},\mathbf{y})^2}$$

Using Lagrange multiplier λ, let $J = (\mathbf{x}-\mathbf{y})^T W(\mathbf{x}-\mathbf{y}) + \lambda(\mathbf{a}^T\mathbf{x}+b)$.

$$\begin{aligned}
\frac{\partial J}{\partial \mathbf{x}} = 0 \Rightarrow \mathbf{x}^* - \mathbf{y} &= -\frac{1}{2}\lambda W^{-1}\mathbf{a} \\
\frac{\partial J}{\partial \lambda} = 0 \Rightarrow \mathbf{a}^T\mathbf{x}^* + b &= 0 \\
\Rightarrow \lambda = \frac{2(\mathbf{a}^T\mathbf{y}+b)}{\mathbf{a}^T W^{-1}\mathbf{a}}, \mathbf{x}^* - \mathbf{y} &= \frac{-(\mathbf{a}^T\mathbf{y}+b)W^{-1}\mathbf{a}}{\mathbf{a}^T W^{-1}\mathbf{a}} \\
\Rightarrow (\mathbf{x}^*-\mathbf{y})^T W(\mathbf{x}^*-\mathbf{y}) &= \frac{(\mathbf{a}^T\mathbf{y}+b)^2}{\mathbf{a}^T W^{-1}\mathbf{a}} \\
\Rightarrow d_W(\mathbf{y},H) = d_W(\mathbf{x}^*,\mathbf{y}) &= \frac{|\mathbf{a}^T\mathbf{y}+b|}{\sqrt{\mathbf{a}^T W^{-1}\mathbf{a}}}
\end{aligned}$$

We note that if W were the identity matrix, then the formula reduces to the known version for Euclidean distance. Next consider two weight matrices W_1 and W_2, it is easy to note that

$$\frac{d_{W_1}(\mathbf{y},H)}{d_{W_2}(\mathbf{y},H)} = \sqrt{\frac{\mathbf{a}^T W_2^{-1}\mathbf{a}}{\mathbf{a}^T W_1^{-1}\mathbf{a}}} \qquad (1)$$

In other words, the ratio of point-to-hyperplane distances under differing weight matrices is *independent* of the point $\mathbf{y}$ (as well as the fixed translation b).

3. ADAPTIVE CLUSTER DISTANCE BOUNDING

It is easy to show that for any positive definite W, the shortest path between two points is along the straight line passing through the two points. Now, given a cluster $\mathcal{X}_m$, the query $\mathbf{q}$ and a hyperplane H that lies between the cluster and the query (a "*separating hyperplane*", see Figure 3), by simple geometry it is easy to see that for any $\mathbf{x} \in \mathcal{X}_m$

$$\begin{aligned}
d_W(\mathbf{q},\mathbf{x}) &\geq d_W(\mathbf{q},H) + d_W(\mathbf{x},H) \\
&\geq d_W(\mathbf{q},H) + \min_{\mathbf{x}\in\mathcal{X}_m} d_W(\mathbf{x},H) \\
&= d_W(\mathbf{q},H) + d_W(\mathcal{X}_m,H) \\
\Rightarrow d_W(\mathbf{q},\mathcal{X}_m) &\geq d_W(\mathbf{q},H) + d_W(\mathcal{X}_m,H) \qquad (2)
\end{aligned}$$

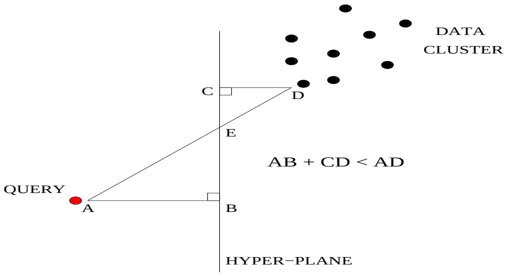

Fig. 2. The Hyperplane Bound

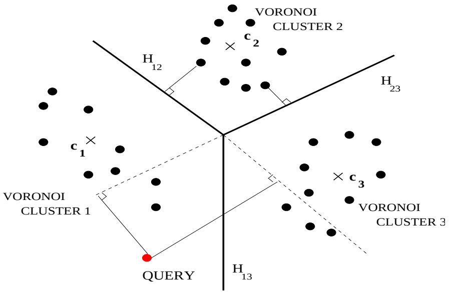

Fig. 3. Cluster Distance Bounding

We focus on the second term, $d_W(\mathcal{X}_m, H)$, the "*support*". Had W been known in advance, this could have been evaluated offline and stored. Instead, let us denote the weight matrix used during clustering as W_0. Then, (1) implies

$$d_W(\mathcal{X}_m, H) = \sqrt{\frac{\mathbf{a}^T W_0^{-1}\mathbf{a}}{\mathbf{a}^T W^{-1}\mathbf{a}}}\, d_{W_0}(\mathcal{X}_m, H) \qquad (3)$$

which demonstrates that it is *unnecessary* to reevaluate the support due to change in weight matrix after the clustering phase. Without loss of generality, in subsequent discussion, we will assume that $d_{W0}(\cdot,\cdot)$ is the Euclidean distance, and drop the suffix W_0.

If $\mathcal{H}_{sep}$ represents a countably finite set of separating hyperplanes (that lie-between the query $\mathbf{q}$ and the cluster $\mathcal{X}_m$),

$$d_W(\mathbf{q}, \mathcal{X}_m) \geq \max_{H \in \mathcal{H}_{sep}} \{d_W(\mathbf{q}, H) + d_W(\mathcal{X}_m, H)\} \qquad (4)$$

The second lower bound presented in (4) can be used to tighten the lower bound on $d_W(\mathbf{q}, \mathcal{X}_m)$. Next, we note that the *boundaries* between clusters generated by the K-means algorithm are *linear hyperplanes*. If $\mathbf{c}_1$ and $\mathbf{c}_2$ are centroids of two clusters $\mathcal{X}_1$ and $\mathcal{X}_2$, and H_{12} the boundary between them, then $\forall \mathbf{y} \in H_{12}$

$$\begin{aligned} d(\mathbf{c}_1, \mathbf{y}) &= d(\mathbf{c}_2, \mathbf{y}) \\ \Rightarrow \|\mathbf{c}_1\|_2^2 - \|\mathbf{c}_2\|_2^2 - 2(\mathbf{c}_1 - \mathbf{c}_2)^T\mathbf{y} &= 0 \end{aligned}$$

Therefore, the hyperplane $H_{12} = H(-2(\mathbf{c}_1 - \mathbf{c}_2), \|\mathbf{c}_1\|_2^2 - \|\mathbf{c}_2\|_2^2)$ is the boundary between the clusters $\mathcal{X}_1$ and $\mathcal{X}_2$. We emphasize that these hyperplane boundaries *need not be stored*, as they can be generated during run-time from the centroids. It is straightforward to show that: Given a query $\mathbf{q}$ and a hyperplane H_{mn} that separates clusters $\mathcal{X}_m$ and $\mathcal{X}_n$, it lies between the query and cluster $\mathbf{X}_m$ ***if and only if*** $d(\mathbf{q}, \mathbf{c}_m) \geq d(\mathbf{q}, \mathbf{c}_n)$.

3.1. Reduced Complexity Hyperplane Bound

For evaluation of the lower-bound of (2) and (4), we would need to pre-calculate and store $d(H_{mn}, \mathcal{X}_m)$ for all cluster pairs (m, n). With K clusters, there are $K(K-1)$ distances that need to be pre-calculated and stored, in addition to the cluster centroids themselves. The total storage for all clusters would be $O(K^2 + Kd)$. This heavy storage overhead makes the hyperplane bound, in this form, impractical for a very large number of clusters. However, we can loosen the bound in (4) as follows:

$$\begin{aligned} d_W(H, \mathcal{X}_m) &= \sqrt{\frac{\|\mathbf{a}\|_2^2}{\mathbf{a}^T W^{-1}\mathbf{a}}} d(\mathcal{X}_m, H)\} \\ &\geq \sqrt{\frac{\|\mathbf{a}\|_2^2}{\mathbf{a}^T W^{-1}\mathbf{a}}} \min_{H \in \mathcal{H}_{sep}} d(\mathcal{X}_m, H)\} \\ \Rightarrow d_W(\mathbf{q}, \mathcal{X}_m) &\geq \max_{\mathcal{H}_{sep}} \{d_W(\mathbf{q}, H) + \sqrt{\frac{\|\mathbf{a}\|_2^2}{\mathbf{a}^T W^{-1}\mathbf{a}}} d_{sep}\} \end{aligned}$$

where $d_{sep} = \min_{H \in \mathcal{H}_{sep}} d(\mathcal{X}_m, H)$. This means that for every cluster $\mathcal{X}_m$ we would only need to store one distance term d_{sep}, thus reducing the total storage to $O(K(d+1))$. For the special case when $d_W(\cdot,\cdot)$ is itself Euclidean, i.e., no weight adaptation, see [8]. For small M, even $\|\mathbf{a}\|_2$, for all cluster boundaries $\mathbf{a}$, can be calculated offline and stored. Even otherwise, we note that it is IO time (and not processor time) which is the bottleneck in query processing.

4. EXPERIMENTAL RESULTS

We compared the performance of our index (henceforth referred to as 'VQ-Hyperplane') with a well-known variant of VA-File [7] that is adapted to leverage relevance feedback. Our data-set BIORETINA[1] consists of MPEG-7 texture feature descriptors extracted from 64 × 64 blocks generated from images of tissue sections of feline retinas as a part of an ongoing project at the Center for Bio-Image Informatics, UCSB. It is 208,506 elements long and 62 dimensional. We also assumed a *page size* of 8kB. The query sets themselves were generated by randomly selecting 100 elements from the relevant data-sets. For each query, the 10 nearest neighbors (10NN) were mined.

The weight matrix, typically a correlation matrix [5], was modelled as $W = U^T \Lambda U$. The orthonormal matrix U was

[1] Available for download from http://scl.ece.ucsb.edu/datasets/features.txt

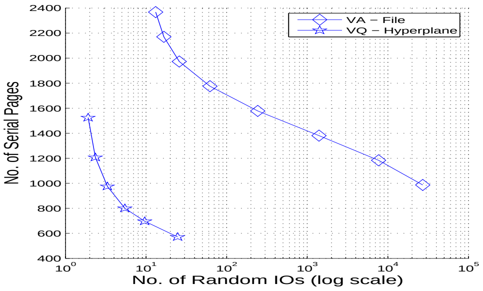

Fig. 4. IO Performance

generated randomly and the eigenvalues were uniformly distributed between 0 and 10. We present results from one such realization of W, that is representative of general performance.

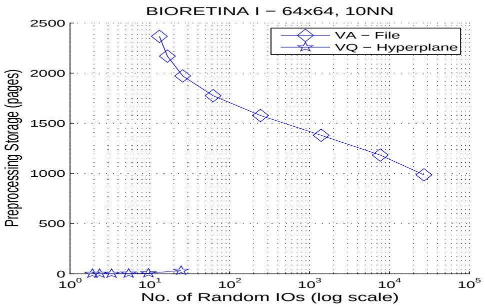

Fig. 5. Preprocessing Storage

We evaluated the performance of VA-File at various quantization levels (5-12 bits per dimension) and the VQ method for varying numbers of clusters (10-300 clusters). We note that our index 'VQ-Hyperplane' is able to consistently reduce the number of random IO reads as compared with VA-File, when allowed (roughly) the same number of sequential disk accesses. For BIO-RETINA (Figure 4), at 6 bit quantization for VA-File, a nearly $3000X$ reduction in costly random disk accesses is achieved by the vector quantization/clustering approach with 15 clusters.

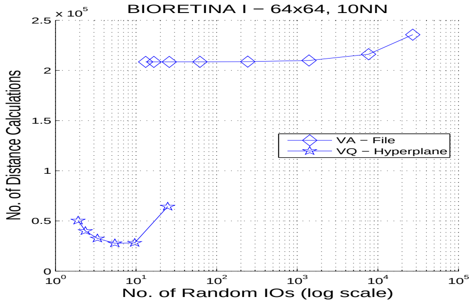

Fig. 6. Computational Cost

Since VA-File maintains a separate compressed representation for each element of the database, the approximation file size grows with the size of the database. Secondly, in order to reduce the number of costly random access reads, the quantization resolution in each dimension needs to be increased, which again results in larger approximation files. In contradistinction, the VQ method reduces random IO reads by *reducing* the number of clusters. Moreover, VA-File estimates distance bounds to each element of the database, not to each cluster as in our method. Hence, we note that the VQ method has significantly ($\approx 10X - 100X$) lower storage and lower computational costs (Figure 5 and 6).

5. CONCLUSIONS

We developed a cluster distance estimation technique that provides tight distance estimates while adapting to changes in the distance metric, and achieves efficient spatial filtering (at low storage and computation costs). The IO access times of our index are significantly lower than VA-File and enables effective application of relevance feedback techniques.

6. REFERENCES

[1] B.S. Manjunath, J.-R. Ohm, V.V. Vasudevan, and A. Yamada, "Color and texture descriptors.," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 703–715, June 2001.

[2] R. Weber, H.J. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces.," in *VLDB*, August 1998, pp. 194–205.

[3] R.E. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, Princeton, NJ, 1961.

[4] T. Huang and X. S. Zhou, "Image retrieval with relevance feedback: From heuristic weight adjustment to optimal learning methods," in *ICIP*, 2001, vol. 3, pp. 2–5.

[5] Y. Ishikawa, R. Subramanya, and C. Faloutsos, "Mindreader: Querying databases through multiple examples," in *VLDB*, August 1998, pp. 218–227.

[6] Y. Rui and T. Huang, "Optimizing learning in image retrieval," *CVPR*, vol. 1, pp. 1236–1243, 2000.

[7] Y. Sakurai, M. Yoshikawa, R. Kataoka, and S. Uemura, "Similarity search for adaptive ellipsoid queries using spatial transformation," in *VLDB*, 2001, pp. 231–240.

[8] S. Ramaswamy and K. Rose, "Adaptive cluster-distance bounding for similarity search in image databases," in *ICIP*, 2007, vol. 6, pp. 381–384.