**Query Response System for File Management**

**Abstract:**

We consider approaches for similarity search in correlated, high-dimensional data sets, which are derived within a clustering framework. We note that indexing by “vector approximation” (VA-File), which was proposed as a technique to combat the “Curse of Dimensionality,” employs scalar quantization, and hence necessarily ignores dependencies across dimensions, which represents a source of sub optimality. Clustering, on the other hand, exploits interdimensional correlations and is thus a more compact representation of the data set. However, existing methods to prune irrelevant clusters are based on bounding hyperspheres and/or bounding rectangles, whose lack of tightness compromises their efficiency in exact nearest neighbor search. We propose a new cluster-adaptive distance bound based on separating hyperplane boundaries of Voronoi clusters to complement our cluster based index. This bound enables efficient spatial filtering, with a relatively small preprocessing storage overhead and is applicable to euclidean and Mahalanobis similarity measures. Experiments in exact nearest-neighbor set retrieval, conducted on real data sets, show that our indexing method is scalable with data set size and data dimensionality and outperforms several recently proposed indexes. Relative to the VA-File, over a wide range of quantization resolutions, it is able to reduce random IO accesses, given (roughly) the same amount of sequential IO operations, by factors reaching 100X and more.

**Existing System:**

However, existing methods to prune irrelevant clusters are based on bounding hyperspheres and/or bounding rectangles, whose lack of tightness compromises their efficiency in exact nearest neighbor search.

Spatial queries, specifically nearest neighbor queries, in high-dimensional spaces have been studied extensively. While several analyses have concluded that the nearest neighbor search, with Euclidean distance metric, is impractical at high dimensions due to the notorious “curse of dimensionality”, others have suggested that this may be over pessimistic. Specifically, the authors of have shown that what Determines the search performance (at least for R-tree-like structures) is the *intrinsic* dimensionality of the data set and not the dimensionality of the address space (or the *embedding* dimensionality).

We extend our distance bounding technique to the Mahalanobis distance metric, and note large gains over existing indexes.

**Proposed System:**

We propose a new cluster-adaptive distance bound based on separating hyperplane boundaries of Voronoi clusters to complement our cluster based index. This bound enables efficient spatial filtering, with a relatively small pre-processing storage overhead and is applicable to Euclidean and Mahalanobis similarity measures. Experiments in exact nearest-neighbor set retrieval, conducted on real data-sets, show that our indexing method is scalable with data-set size and data dimensionality and outperforms several recently proposed indexes.

we outline our approach to indexing real high-dimensional data-sets. We focus on the clustering paradigm for search and retrieval. The data-set is clustered, so that clusters can be retrieved in decreasing order of their probability of containing entries relevant to the query.

We note that the Vector Approximation (VA)-file technique implicitly assumes independence across dimensions, and that each component is uniformly distributed. This is an unrealistic assumption for real data-sets that typically exhibit significant correlations across dimensions and non-uniform distributions. To approach optimality, an indexing technique must take these properties into account. We resort to a Voronoi clustering framework as it can naturally exploit correlations across dimensions (in fact, such clustering algorithms are the method of choice in the design of vector quantizers). Moreover, we show how our clustering procedure can be combined with any other generic clustering method of choice (such as BIRCH ) requiring only one additional scan of the data-set. Lastly, we note that the sequential scan is in fact a special case of clustering based index i.e. with only one cluster.

Several index structures exist that facilitate search and retrieval of multi-dimensional data. In low dimensional spaces, *recursive* partitioning of the space with hyper-rectangles hyper-spheres or a combination of hyper-spheres and hyper-rectangles have been found to be effective for nearest neighbor search and retrieval. While the preceding methods specialize to Euclidean distance (*l*2 norm), M-trees have been found to be effective for metric spaces with arbitrary distance functions (which are metrics).

Such multi-dimensional indexes work well in low dimensional spaces, where they outperform sequential scan. But it has been observed that the performance degrades with increase in feature dimensions and, after a certain dimension threshold, becomes inferior to sequential scan. In a celebrated result,

Weber et. Al have shown that whenever the dimensionality is above 10, these methods are outperformed by simple sequential scan. Such performance degradation is attributed to Bellman’s ‘*curse of dimensionality*’, which refers to the exponential growth of hyper-volume with dimensionality of the space.

**Module Description:**

1. **A New Cluster Distance Bound**
2. **Adaptability to Weighted Euclidean or Mahalanobis Distances**
3. **An Efficient Search Index**
4. **Vector Approximation Files**
5. **Approximate Similarity Search**

**A New Cluster Distance Bound**

Crucial to the effectiveness of the clustering-based search strategy is efficient bounding of query-cluster distances. This is the mechanism that allows the elimination of irrelevant clusters. Traditionally, this has been performed with bounding spheres and rectangles. However, hyperspheres and hyperrectangles are generally not optimal bounding surfaces for clusters in high dimensional spaces. In fact, this is a phenomenon observed in the SR-tree, where the authors have used a combination spheres and rectangles, to outperform indexes using only bounding spheres (like the SS-tree) or bounding rectangles (R*∗*-tree).

The premise herein is that, at high dimensions, considerable improvement in efficiency can be achieved by relaxing restrictions on the regularity of bounding surfaces (i.e., spheres or rectangles). Specifically, by creating Voronoi clusters, withpiecewise-linear boundaries, we allow for more general convex polygon structures that are able to efficiently bound the cluster surface. With the construction of Voronoi clusters under the Euclidean distance measure, this is possible. By projection onto these hyperplane boundaries and complementing with the cluster-hyperplane distance, we develop an appropriate lower bound on the distance of a query to a cluster.

**Adaptability to Weighted Euclidean or Mahalanobis Distances**

While the Euclidean distance metric is popular within the multimedia indexing community it is by no means the “correct” distance measure, in that it may be a poor approximation of user perceived similarities. 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 and more recently) . We extend our distance bounding technique to the Mahalanobis distance metric, and note large gains over existing indexes.

**An Efficient Search Index**

The data set is partitioned into multiple Voronoi clusters and for any kNN query, the clusters are ranked in order of the hyperplane bounds and in this way, the irrelevant clusters are filtered out. We note that the sequential scan is a special case of our indexing, if there were only one cluster. An important feature of our search index is that we do not store the hyperplane boundaries (which form the faces of the bounding polygons), but rather generate them dynamically, from the cluster centroids. The only storage apart from the centroids are the *cluster-hyperplane boundary distances* (or the *smallest* cluster-hyperplane distance). Since our bound is relatively tight, our search algorithm is effective in spatial filtering of

irrelevant clusters, resulting in significant performance gains. We expand on the results and techniques initially presented in , with comparison against several recently proposed indexing techniques.

**Vector Approximation Files**

A popular and effective technique to overcome the curse of dimensionality is the vector approximation file (VA-File). VA-File partitions the space into hyper-rectangular cells, to obtain a quantized approximation for the data that reside inside the cells. Non-empty cell locations are encoded into bit strings and stored in a separate *approximation file*, on the hard-disk. During a nearest neighbor search, the vector approximation file is sequentially scanned and upper and lower bounds on the distance from the query vector to each cell are estimated. The bounds are used to prune irrelevant cells. The final set of candidate vectors are then read from the hard disk and the exact nearest neighbors are determined. At this point, we note that the terminology “Vector Approximation” is somewhat confusing, since what is actually being performed is *scalar quantization*, where each component of the feature vectors *separately and uniformly quantized* (in contradistinction with vector quantization in the signal compression literature).

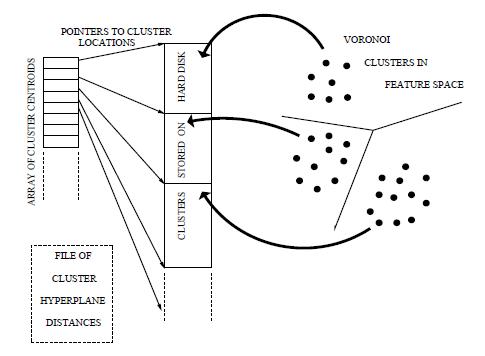
VA-File was followed by several more recent techniques to overcome the curse of dimensionality. In the VA+-File, the data-set is rotated into a set of uncorrelated dimensions, with more approximation bits being provided for dimensions with higher variance. The approximation cells are adaptively spaced according to the data distribution. Methods such as LDR and the recently proposed non-linear approximations aim to outperform sequential scan by a combination of clustering and dimensionality reduction. There also exist a few hybrid methods, such as the A-Tree, and IQ-Tree, which combine VA-style approximations within a tree based index.

**Approximate Similarity Search**

Lastly, it has been argued that the feature vectors and distance functions are often only approximations of user perception of similarity. Hence, even the results of an exact similarity search is inevitably *perceptually* approximate, with additional rounds of query refinement necessary. Conversely, by performing an approximate search, for a small penalty in

accuracy, considerable savings in query processing time would be possible. Examples of such search strategies are MMDR probabilistic searches and locality sensitive hashing .The reader is directed to for a more detailed survey of approximate similarity search. The limits of approximate indexing i.e. the optimal tradeoffs between search quality and search time has also been studied within an information theoretic framework.

**System Architecture:**

****

# Hardware System Requirement

# Processor - Pentium –III

Speed - 1.1 Ghz

RAM - 256 MB(min)

Hard Disk - 20 GB

Floppy Drive - 1.44 MB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - SVGA

# S/W System Requirement

* **Operating System :**Windows 95/98/2000/NT4.0.
* **Application Server : Tomcat6.0**
* **Front End :** HTML, Java.
* **Scripts :** JavaScript.
* **Server side Script :** Java Server Pages.
* **Database :** Mysql.
* **Database Connectivity :** JDBC.