FILTERING OF HIGH DIMENSIONAL DATA FOR IMPROVING THE SIMILARITY SEARCH CORRELATED DATA USING SPATIAL FILTERING

[1]Ancy S, [2]K.Cornelius
[1]Assistant Professor, [2]Assistant Professor,
[1]Department of Information Technology, [2]Department of Information Technology,
[1]Jeppiaar Institute of Technology, [2]St.Peters College of Engineering and Technology.
Chennai, Tamilnadu
[1]sancyit@gmail.com,[2]cornelius851@gmail.com

Abstract: We consider approaches for similarity search in correlated, high- dimensional data sets, which are derived within a clustering framework. Indexing by "vector approximation" (VA-File), which was proposed as a technique to combat the "Curse of Dimensionality," employs scalar quantization. It ignores dependencies across dimensions, which represents a source of suboptimality. Clustering, on the other hand, exploits interdimensional correlations and is thus a more compact representation of the data set. We developed cluster distance bounds based on separating hyperplane boundaries and our search index, complemented by these bounds, is applicable to Euclidean and Mahalanobis distance metrics. It obtained significant reductions in number of random IOs over several recently proposed indexes, when allowed (roughly) the same number of sequential pages, has a low computational cost and scales well with dimensions and size of the data-set. We note that while the hyperplane bounds are better than MBR and MBS bounds, they are still loose compared with the true query-cluster distance.

Keywords: **R* tree generation, Indexing, hyper plane boundary, prune clusters.**

Introduction:

This has spawned new applications such as Multimedia information Systems, CAD/CAM, Geographical Information systems (GIS), medical imaging, time-series analysis (in stock markets and sensors), that store large amounts of data periodically in and later, retrieve it from databases. The size of these databases can range from the relatively small (a few 100 GB) to the very large (several 100TB, or more).In the future, large organizations will have to retrieve and process petabytes of data, for various purposes such as data mining and decision support. Thus, there exist numerous applications that access large multimedia databases, which need to be effectively supported.

Existing System:

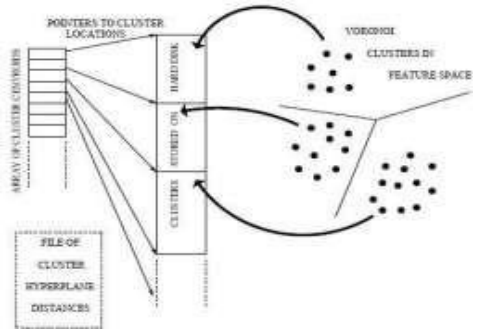
Existing methods to prune irrelevant clusters are based on bounding hyperspheres and/or bounding rectangles, whose lack of tightness compromises their efficiency in exact nearest neighbor search. Spatial queries, specifically nearest neighbor queries, in high- dimensional spaces have been studied extensively. Euclidean distance metric, is impractical at high dimensions due to the notorious "curse of dimensionality".

Issues:

Preprocessing storage is not supported in the existing system. Time consuming process to extract the data.

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 out performs several recently proposed indexes



Implementation:

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.

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. Since our bound is relatively tight, our search algorithm is effective in spatial filtering of irrelevant clusters, resulting in significant performance gains.

Performance Measure:

The common performance metrics for exact nearest neighbour search have been to count page accesses or the response time. However, page accesses may involve both serial disk accesses and random IOs, which have different costs. On the other hand, response time (seek times and latencies) is tied to the hardware being used and therefore, the performance gain/loss would be platform dependent. In the VA-File the number of quantization levels per dimension can be varied.

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. In fact, this is a phenomenon observed in the SR-tree. 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 Mahalonobis Distances:

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

Related Work:

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(Rtrees,R* -trees), hyper- spheres (SS-Tree or a combination of hyper-spheres and hyper-rectangles (SR-Tree), have been found to be effective for nearest neighbour search and retrieval. Such multidimensional indexes work well in low dimensional spaces, where they outperform sequential scan

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 obtained 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 approximation cells are adaptively spaced according to the data distribution

Approximate Similarity Search:

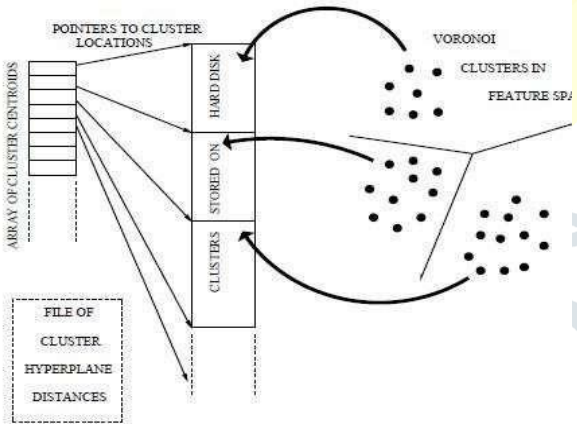
It has been argued that the feature vectors and distance functions are often only approximations of user perception of similarity. By performing an approximate search, for a small penalty in accuracy, considerable savings in query processing time would be possible. The optimal tradeoffs between search quality and search time has also been studied within an information theoretic framework.

Clustering And Index Structure:

The first step in index construction is the creation of Nearest Neighbour/Voronoi clusters There exist several techniques of clustering the data-set, from the fast K-means algorithm. We believe the centroid is a good choice as a pivot. Thus, quick Voronoi clustering, with possibly only a single scan of the entire data-set, can be achieved using any generic clustering algorithm. Faster index construction would be possible by allowing for hierarchical and multi-stage clustering.

Storage Statergy:

Elements within the same cluster are stored together. We retain the cluster centroids cm and maintain pointers from each centroid to the location of the corresponding cluster on the hard- disk. We also maintain in a separate file the distance of each cluster from its bounding hyperplanes. we note that the hyperplane boundaries can be generated in run-time and assume systems of sufficient mainmemory capacity to allow storage of intermediate results.



The Knn Search Algorithm:

We now present KNN- SEARCH, our procedure for k-NN search. Our algorithm is a branch- and-bound algorithm, and since the clusters are accessed in order of the lower bounds to the query distance, it is guaranteed to return the k-nearest neighbours.

Conclusion:

We proposed an indexing method,based upon principles of vector quantization,where the data set is partitioned into voronoi clusters and clusters are accessed in order of the query cluster distances. We developed cluster distance bounds based on seperating hyperplane boundaries and our search index, complemented by these bounds, is applicable to Mahalanobis distance metrics. Conceivably, the cluster-distance bounds can be further tightened.

References:

[1] V. Cardellini, M. Colajanni, and S. Yu. Dynamic load balancing on web-server systems. IEEE Internet Computing, 3(3):28{39, 1999. [2] L. Cherkasova. FLEX: Load Balancing and Management Strategy for Scalable Web Hosting Service. IEEE Symposium on Computers and Communications, 0:8, 2000.
[3] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, Leach, and T. Berners-Lee. Hypertext transfer protocol { http/1.1. In IETF RFC 2616, 1999.
[4] G. Hunt, E. Nahum, and J. Tracey. Enabling content- basedload distribution for scalable services. Technical report, 1997.
[5] E. Katz, M. Butler, and R. McGrath. A scalable HTTP server: The NCSA prototype. In Proc. First International Conference on the World Wide We
[6] b, Apr. 1994

FILTERING OF HIGH DIMENSIONAL DATA FOR IMPROVING THE SIMILARITY SEARCH CORRELATED DATA USING SPATIAL FILTERING

[1]Ancy S, [2]K.Cornelius
[1]Assistant Professor, [2]Assistant Professor,
[1]Department of Information Technology, [2]Department of Information Technology,
[1]Jeppiaar Institute of Technology, [2]St.Peters College of Engineering and Technology.
Chennai, Tamilnadu
[1]sancyit@gmail.com,[2]cornelius851@gmail.com