Vector Approximation File: Cluster Bounding in High-Dimension Data Set

Poonam Yerpude

Abstract*— In many modern application ranges high-dimensional feature vectors are used to model complex data sets. We have proposed an overview about efficient indexing method for high-dimensional database using an filtering approach known as vector approximation approach which supports the nearest neighbor search efficiently And A cluster distance bound based on separating hyper planes, that complements our index in electively retrieving clusters that contain data entries closest to the query. The Creation of approximation for Vectors for use in similarity (also known the retrieval of k-nearest neighbor) is examined.

Index Terms*— Similarity Search, indexing, vector quantization, clustering, Nearest Neighbor search.

I. INTRODUCTION

Many emerging database applications such as image, time series and scientific databases, manipulate high dimensional data. In these applications, one of the most frequently used and yet expensive operations is to find objects in the high-dimensional database that are similar to a given query object. Nearest neighbor search is a central requirement in such cases. There is a long stream of research on solving the nearest neighbor search problem, and a large number of multidimensional indexes have been Developed for this purpose. However, most of these structures are not adaptive with respect to data distribution. In consequence, they tend to perform well for some data sets and poorly for others.

In some applications, such as GIS, the feature vectors usually have small number of dimensions, typically 2 dimensions. Numerous index structures exist that facilitate search and retrieval of two dimensional data or spatial databases [1,3]. The general approach for high dimensional indexing was to extend the spatial index structures and to propose new ones to deal with the high dimensional nature of information. Weber et al. [4] have developed a quantitative analysis and performance study of similarity search techniques for high dimensional data sets. They formally show that for data sets with uniform distribution, the indexing techniques based on partitioning and clustering are outperformed on average by a simple sequential scan if the number of dimensions exceeds around 10. Under the assumption of uniformity and independence, they formally establish that there is no organization of high dimensional vector space based on partitioning or clustering which does not degenerate to a sequential scan if dimensionality exceeds a certain threshold. Instead of the data space or data partitioning, or clustering based approaches which suffer from the dimensionality curse, a new approach based on vector approximations is proposed which overcomes these problems and scales well even if dimensionality increases.

The technique proposed in [4] is called VA File, i.e., vector approximation file, and the technique has been shown to be very effective compared to current indexing techniques. The biggest advantage of the technique is that it does not suffer from the dimensionality problem. Although it provides significant improvements compared to current techniques, the VA-file itself suffers from various problems. In this paper, we discuss these problems and propose a solution that renders the vector approximation idea more effectively. The major problems in the current VA-file that need to be overcome can be summarized as the assumption of independent or uncorrelated dimensions, uniform bit allocation, and the simple partitioning technique. The dimensions of the feature vectors in a real data set usually are dependent or correlated. Moreover, real data sets are not uniformly distributed. More careful analysis for non-uniformed correlated data is needed for effectively indexing high dimensional data. We discuss the problems that the VA-file suffers from if the data set is not uniformly distributed, especially if it is highly correlated or clustered.

II. DEFINITION

Substantial activity was devoted this year to the problem of exact nearest-neighbor search in high-dimensional spaces. The Vector Approximation File (VA-File) approach is in fact based on uniform scalar quantization of feature vectors, and is a powerful technique that scales well with size and dimensionality of the data-set. Since feature vectors often exhibit correlations and dependencies across dimensions, one should expect a search technique based on vector quantization to achieve performance gains. Building on our earlier findings that clustering-based search methods are inherently advantageous as the overall framework within which one incorporates compression-based search techniques, and noting that the K-means clustering algorithm is virtually identical to the Generalized Lloyd Algorithm from vector quantization, we designed an indexing technique that competes with VA-File by accurately identifying and retrieving only the necessary clusters.

Manuscript Details: December 2011

Poonam Yerpude, Computer Science Department and Engg. RCET Bhiali, CSVTU Raipur, India, MobileNo. 9993649790, E-mail: poonamyerpude@gmail.com

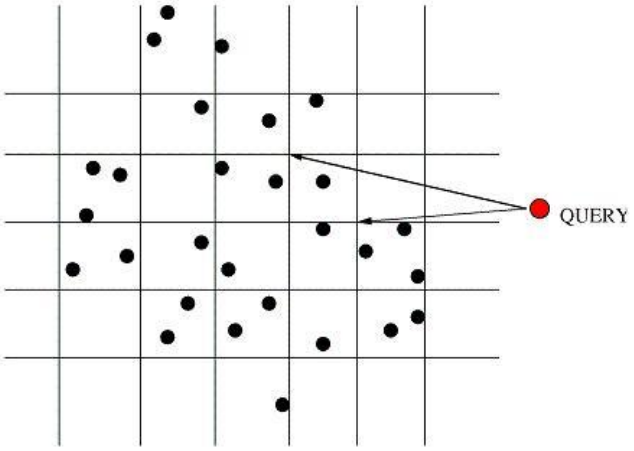

Fig 1: ***Vector Approximation (VA) –File***

III. VECTOR APPROXIMATION FILES

In high dimensional spaces tree indexing structures become useless because an increasing percentage of the nodes need to be examined anyway. To speed up linear search, a compressed version of the feature vectors stored in RAM is used to prefilled the datasets in a first run. The final candidates are determined in a second stage using the uncompressed data from the disk for distance calculation. A popular and effective technique to overcome the curse of dimensionality is the vector approximation file (VA-File) [8]. 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 neighbour 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 neighbours 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 vector is *separately and uniformly quantized* (in contradistinction with vector quantization in the signal compression literature).

3.1 Indexing Based on Vector Approximation

The VA-file approach divides the data space into 2b rectangular cells where b is the total number of bits specified by the user [4]. Each dimension is allocated a number of bits, which are used to divide it into equal populated cells on that dimension. Each cell has a bit representation of length b which approximates the data points that fall into a cell by the corresponding bit representation of the cell. The VA-file itself is simply an array of these bit vector approximations based on the quantization of the original feature vectors.

3.2 Vector quantization

Vector quantization is a classical quantization technique from signal processing which allows the modeling of probability density functions by the distribution of prototype vectors. It was originally used for data compression. It works by dividing a large set of points (vectors) into groups having approximately the same number of points closest to them. Each group is represented by its centroid point, as in k-means and some other clustering algorithms.

The density matching property of vector quantization is powerful, especially for identifying the density of large and high-dimensioned data. Since data points are represented by the index of their closest centroid, commonly occurring data have low error, and rare data high error. This is why VQ is suitable for lossy data compression. It can also be used for lossy data correction and density estimation.

- *Training*

A simple training algorithm for vector quantization is:

1. Pick a sample point at random
2. Move the nearest quantization vector centroid towards this sample point, by a small fraction of the distance
3. Repeat

A more sophisticated algorithm reduces the bias in the density matching estimation, and ensures that all points are used, by including an extra sensitivity parameter:

1. Increase each centroid's sensitivity by a small amount
2. Pick a sample point at random
3. Find the quantization vector centroid with the smallest <distance-sensitivity>

Move the chosen centroid toward the sample point by a small fraction of the distance

Set the chosen centroid's sensitivity to zero

4. Repeat

The algorithm can be iteratively updated with Complex data, rather than by picking random points from a data set, but this will introduce some bias if the data is temporally correlated over many samples.

IV. CLUSTERING

Clustering high-dimensional data is the cluster analysis of data with anywhere from a few dozen to many thousands of dimensions. Cluster analysis or clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters.

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with low distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery that involves try and failure. It will often be necessary to modify pre-processing and parameters until the result achieves the desired properties.

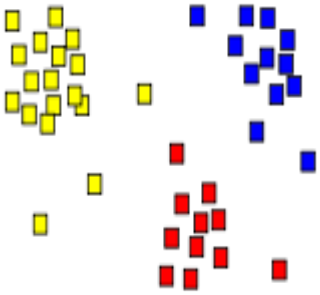

Fig.2 The result of a cluster analysis shown as the coloring of the squares into three clusters.

4.1Correlation clustering

Correlation clustering also relates to a different task, where correlations among attributes of feature vectors in a high-dimensional space are assumed to exist guiding the clustering process. These correlations may be different in different clusters, thus a global de correlation cannot reduce this to traditional (uncorrelated) clustering.

Correlations among subsets of attributes result in different spatial shapes of clusters. Hence, the similarity between cluster objects is defined by taking into account the local correlation patterns. With this notion, the term has been introduced in [7] simultaneously with the notion discussed above. Different methods for correlation clustering of this type are discussed in [7]the relationship to different types of clustering is discussed in[6], see also Clustering high-dimensional data.

Correlation clustering (according to this definition) can be shown to be closely related to biclustering. As in biclustering, the goal is to identify groups of objects that share a correlation in some of their attributes; where the correlation is usually typical for the individual clusters.

The VA-File approach is a special case of a compression based search, where each feature component is compressed uniformly and independently. The optimal compression technique in multidimensional spaces is Vector Quantization (VQ), implemented through clustering. The database is clustered and the most "promising" clusters are retrieved. Huge-gains over VA-File, tree-based indexes and sequential scan have been observed [5,6].

The Hyper plane Bound

Central to this index is adaptive bounding of the query-cluster distances. We investigated the effectiveness of known distance bounding methods namely those based on bounding hyper-rectangles, rotated bounding hyper-rectangles and bounding hyper spheres. This led to and motivated our development of a new and superior bounding technique that is based on distance projections onto separating hyper planes between clusters.

Centroid-based clustering

In centroid-based clustering, clusters are represented by a central vector, which may not necessarily be a member of the data set. When the number of clusters is fixed to k, *k*-means clustering gives a formal definition as an optimization problem: find the *k* cluster centres and assign the objects to the nearest cluster centre, such that the squared distances from the cluster are minimized.

V. K-NEAREST NEIGHBOURS

The *k*-nearest-neighbour method was first described in the early 1950s. The method is labour intensive when given large training sets, and did not gain popularity until the 1960s when increased computing power became available. It has since been widely used in the area of pattern recognition. A k-nearest neighbor (KNN) query finds the nearest k objects to a query points identifies the top *k* nearest neighbors' to the query. This technique is commonly used in predictive analytics to estimate or classify a point based on the consensus of its neighbors. *K-nearest* neighbor graphs are graphs in which every point is connected to its *k* nearest neighbors.

5.1 Approximate nearest neighbour

In some applications it may be acceptable to retrieve a "good guess" of the nearest neighbour. In those cases, we can use an algorithm which doesn't guarantee to return the actual nearest neighbour in every case, in return for improved speed or memory savings. Often such an algorithm will find the nearest neighbour in a majority of cases, but this depends strongly on the dataset being queried. Algorithms that support the approximate nearest neighbour search include locality-sensitive hashing, best bin first and balanced box-decomposition tree based search[7].

5.2 Nearest neighbour distance ratio

Nearest neighbour distance ratio do not apply the threshold on the direct distance from the original point to the challenger neighbour but on a ratio of it depending on the distance to the previous neighbour. It is used in CBIR to retrieve pictures through a "query by example" using the similarity between local features. More generally it is involved in several matching problems.

All nearest neighbors

For some applications (e.g. entropy estimation), we may have *N* data-points and wish to know which is the nearest neighbour *for every one of those N points*. This could of course be achieved by running a nearest-neighbour search once for every point, but an improved strategy would be an algorithm that exploits the information redundancy between these *N* queries to produce a more efficient search. As a simple example: when we find the distance from point *X* to point *Y*, that also tells us the distance from point *Y* to point *X*, so the same calculation can be reused in two different queries.

5.3 K-nearest-neighbour (kNN)

Classification is one of the most fundamental and simple classification methods and should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution of the data. K-nearest-neighbour classification was developed from the need to perform discriminate analysis when reliable parametric estimates of probability densities are unknown or difficult to determine.

1) Between-sample geometric distance

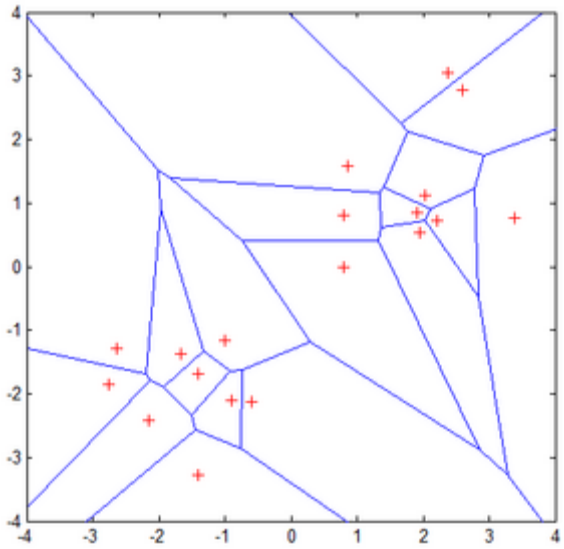

Figure 3: Voronoi tessellation showing Voronoi cells of 19 samples marked with a "+". The Voronoi tessellation reflects two characteristics of the example 2-dimensional coordinate system: i) all possible points within a sample's Voronoi cell are the nearest neighbouring points for that sample, and ii) for any sample, the nearest sample is determined by the closest Voronoi cell edge.

VI. CONCLUSION

In High dimensional data-sets exhibit significant correlations and non-uniform distributions. Hence, indexing with the VA-File, by performing uniform, scalar quantization, is suboptimal. We proposed an indexing method, based upon principles of vector quantization instead, where the data set is partitioned into Voronoi clusters and clusters are accessed in order of the query-cluster distances. Conceivably, the cluster-distance bounds can be further tightened, possibly by optimizing the clustering algorithm so as to optimize the cluster distance bounds. The current VA-file approach assumes independent or uncorrelated dimensions, applies uniform bit allocation, and is based on a simple partitioning technique. More careful analysis for non-uniform or correlated data is needed for effectively indexing real data sets. Real data sets are not uniformly distributed are often clustered, and the dimensions of the feature vectors in real data sets are usually correlated.

REFRENCES

1. Kriegel H.-P. Berchtold S. S3: Similarity search in cad database systems. In Proc. ACM SIGMOD Int. Conf. on Management of Data, pages 564{567, Tuscon, Arizona, 1997.
2. P. Bernstein, M. Brodie, S. Ceri, D. DeWitt,M. Franklin, H. Garcia-Molina, J. Gray, J. Held,J. Hellerstein, H. Jagadish, M. Lesk, D. Maier,J. Naughton, H. Pirahesh, M. Stonebraker, and J. Ull-man. The Asilomar report on database research. ACM Sigmod Record, 27(4), December 1998.
3. S. Berchtold, D. A. Keim, and H. P. Kreigel. The X-tree: An index structure for high-dimensional data. In 22nd. Conference on Very Large Databases, pages 28{39, Bombay, India, 1996.
4. R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for [4]similarity-search methods in high-dimensional spaces. In Proceedings of the Int. Conf. on Very Large Data Bases, pages 194{205,New York City, New York, August 1998.
5. H. Samet. The Design and Analysis of Spatial Structures. Addison Wesley Publishing Company, Inc., Massachusetts, 1989.
6. S. Berchtold, C. Bohm, D. Keim, and H. Kriegel. A cost model for nearest neighbor search in high-dimensional data space. In Proc. ACM Symp. on Principles of Database Systems, pages 78{86, Tuscon, Arizona, June 1997.
7. V. Gaede and O. Gunther. multidimensional access methods. ACM Computing Surveys, 30:170{231, 1998.
8. R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In Proceedings of the Int. Conf. on Very Large Data Bases, pages 194{205,New York City, New York, August 1998.
9. Zimek, A. (2008). *Correlation Clustering*. http://edoc.ub.uni-muenchen.de/8736/.
10. Kriegel, H.-P.; Kröger, P., Zimek, A. (March 2009). "Clustering High Dimensional Data: A survey on Subspace Clustering, Pattern-based Clustering, and Correlation Clustering". *ACM Transactions on Knowledge Discovery from Data (TKDD)* **3** (1): 1–58.:10.1145/1497577.1497578. http://doi.acm.org/10.1145/1497577.1497578
11. Böhm, C., Kailing, K., Kröger, P., Zimek, A. (2004). "Computing Clusters of Correlation Connected Objects". *Proc. ACM SIGMOD International Conference on Management of Data (SIGMOD'04), Paris, France*. pp. 455–467.:10.1145/1007568.1007620. http://doi.acm.org/10.1145/1007568.1007620.
12. R. Weber, H. 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.

13. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression* Kluwer Academic Publishers, 1992.
14. T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases." in *SIGMOD*, 1996, pp. 103–114.
15. K. Chakrabarti and S. Mehrotra. The hybrid tree: An index structure for high dimensional feature spaces. In Proc. Int. Conf. Data Engineering, pages 440{447, Sydney, Australia, 1999.

AUTHORS PROFILE

Poonam Yerpude MTech Computer Technology 2012,CSVTU Bhilai,BE Computer Science and Engg.2008,1-National paper, Data Mining.