Approximate Similarity Search in Metric Data by Using Region Proximity

Giuseppe Amato[1], Fausto Rabitti[2], Pasquale Savino[1], and Pavel Zezula[3]

[1] IEI-CNR, Pisa, Italy,
{G.Amato,P.Savino}@iei.pi.cnr.it
WWW home page: http://www.iei.pi.cnr.it
[2] CNUCE-CNR, Pisa, Italy,
F.Rabitti@cnuce.cnr.it
WWW home page: http://www.cnuce.cnr.it
[3] Masaryk University, Brno, Czech Republic,
zezula@fi.muni.cz
WWW home page: http://www.fi.muni.cz

Abstract. The problem of approximated similarity search for the range and nearest neighbor queries is investigated for generic metric spaces. The search speedup is achieved by ignoring data regions with a small, user defined, proximity with respect to the query. For zero proximity, exact similarity search is performed. The problem of proximity of metric regions is explained and a probabilistic approach is applied. Approximated algorithms use a small amount of auxiliary data that can easily be maintained in main memory. The idea is implemented in a metric tree environment and experimentally evaluated on real-life files using specific performance measures. Improvements of two orders of magnitude can be achieved for moderately approximated search results. It is also demonstrated that the precision of data regions' proximity measure significantly influence approximated algorithms.

1 Introduction

Contrary to traditional databases, where simple attribute data are used, the standard approach to searching modern data repositories, such as multimedia databases, is to perform search on characteristic features that are extracted from information objects. Features are typically high dimensional vectors or some other data items the pairs of which can only be compared by specific functions. In such search environments, *exact match* has little meaning thus concepts of *similarity* are typically applied.

Similarity searching has become fundamental in a variety of application areas, including multimedia information retrieval, data mining, pattern recognition, machine learning, computer vision, genome databases, data compression, and statistical data analysis. This problem was originally studied within the area of *computational geometry*, where it is known as the *closest point*, *nearest neighbor*, or *post office* problem. However, it has recently attracted much attention in the database community, because of the increasingly growing needs to deal with large volume of data. Consequently, *efficiency* has become a matter of concern in prosperous designs. Though a lot of work has been done to develop structures able to perform similarity search fast, results are still not satisfactory, and much more research is needed.

2 The proposed approach

Consider a typical similarity query: find all data objects whose distance from the query object is smaller than a certain threshold. The query object and the threshold form the so called

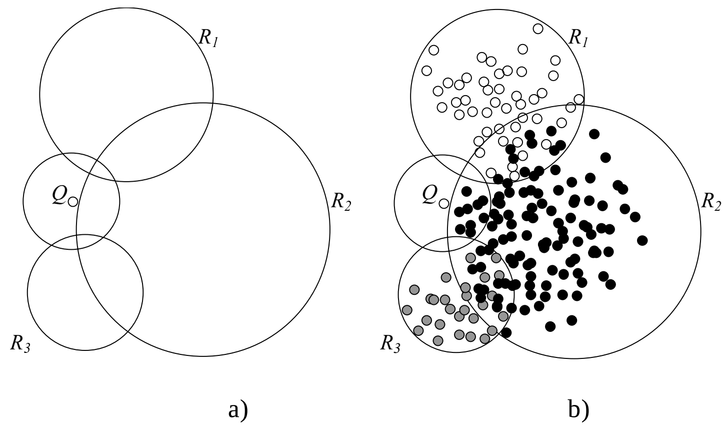

Fig. 1. Query region intersects data regions

query region and the qualifying objects are those that are contained in (or covered by) this region. In order to increase efficiency, index structures divide searched data sets into partitions, and bound such partitions in regions [Gu84,BKK96,BO97,CPZ97,BO99,TTS+00]. Qualifying objects are found accessing only those partitions the regions of which intersect the query region.

However, index structures typically suffer from the so called *dimensionality curse*. It has been observed that, when the number of dimensions of a data set is greater than 10-15, performance of index structures decrease and a linear scan over the whole data set would perform better [BGR+99,WSB98].

One consequence of the dimensionality curse is that the probability of overlaps between the query and data regions is very high. Therefore, the execution of a similarity query may require to access many of the data regions – all data regions that overlap the query region must be accessed – loosing the advantage of any indexing structure what so ever. For instance, in Figure 1a, the query region overlaps regions R_1, R_2 and R_3 so all of them should be accessed to answer to the query. Given this problem and the intrinsically interactive nature of the similarity-based search process, where the efficient execution of elementary queries has become even more important, the notion of *approximate search* [AMN+98,ZSA+98,PAL99,CP00] has emerged as important research issue.

In this article we propose a new technique for approximate similarity search, which is based on some the following observation. In many cases, even if the query region overlaps a data region, no data objects are contained in the intersection. This naturally depends on the data object distribution. The result of this phenomenon is that even though all data regions are accessed, very few of them contain qualifying data objects (very few of them have a non empty intersection with the query region), so many accesses are actually void. In Figure 1b, it can be seen that, although the query region intersects regions R_1, R_2 and R_3, the intersection with R_1 and R_3 is empty so it was not necessary to access them. In the figure, white points correspond to the partition bounded by R_1, black points to the partition bounded by R_2, and gray points to the partition bounded by R_3. There is no way to precisely determine if the intersection between the query region and a data region is empty without accessing the data region itself. However, not accessing *empty* regions would really increase the performance of similarity search algorithms.

Our idea is to use some statistical information to determine the probability that the query region and a data region share some data objects. Of course it could happen that some regions that contain qualifying objects are discarded by our algorithm, so the result is an approximation of the exact result. In our proposal, approximation is controlled by the concept of *proximity of regions* (see Section 3 for a formal definition) that allows to estimate the probability that two regions share data objects. Only data regions whose proximity with the query region is greater than a specified threshold parameter will be accessed. When the parameter is zero, precise results are guaranteed, and the higher the proximity threshold, the less accurate the results are, but the faster the query is executed.

We apply this idea for the similarity range and the nearest neighbors queries and verify its validity on real-life data sets. We consider generic metric spaces that include the n-dimensional vector spaces as special cases. There are three main reasons for considering such environment. First, an increasing number of applications use feature spaces that are not constrained by other properties but the metric postulates (e.g. *Levenstein distance*, *Hausdorff distance*, or *quadratic form distance*). Second, many specialized solutions on vector spaces do not perform sufficiently well even in restricted environments since they are always affected by the dimensionality curse. Third, provided that a good solution for general metric spaces is found, it would work for a large number of existing and possible future distance measures.

3 Definition of proximity

A *metric space* $\mathcal{M} = (\mathcal{D}, d)$ is defined by a domain of objects $\mathcal{D}$ (i.e. the *keys* or indexed *features*) and by a total (distance) function d a *non negative* and *symmetric* function that satisfies the *triangle inequality* property, i.e. $d(O_x, O_y) \leq d(O_x, O_z) + d(O_z, O_y), \forall O_x, O_y, O_z \in \mathcal{D}$. We assume that the maximum distance never exceeds d_m, thus we consider a *bounded metric space*. Given the metric space $\mathcal{M}$, a *ball region* $\mathcal{B}_x = \mathcal{B}_x(O_x, r_x) = \{O_i \in \mathcal{D} \mid d(O_x, O_i) \leq r_x\}$ is determined by a center $O_x \in \mathcal{D}$ and a radius $r_x \geq 0$. It is composed of such objects in $\mathcal{D}$ for which the distance to O_x is less than or equal to r_x.

The proximity $X(\mathcal{B}_x, \mathcal{B}_y)$ of ball regions $\mathcal{B}_x, \mathcal{B}_y$ is the probability that a randomly chosen object $\mathbf{O}$ over the same metric space $\mathcal{M}$ appears in both regions, i.e.

$$X(\mathcal{B}_x, \mathcal{B}_y) = \mathtt{Pr}\{d(\mathbf{O}, O_x) \leq r_x \wedge d(\mathbf{O}, O_y) \leq r_y\} \tag{1}$$

Notice that the proximity cannot be quantified by the amount of space which appear in regions' intersection, because in general metric spaces such quantity cannot be determined.

Recent studies, [ARS$^+$00a,ARS$^+$00b], have proposed some approaches to obtain an accurate and efficient evaluation of proximity.

4 Experimental Evaluation

In this section, we describe the data sets that we used for the experiments, we define the measures to evaluate the approximated search process, and we discuss obtained results. We also make clear through experiments the advantage of applying the probabilistic approximation measure instead of the trivial. Special observations are finally discussed.

4.1 Data Sets

In our experiments, we used color features of two image collections of size 10 000 objects each. In the first data set, designated as HV1, color features are represented as 9-dimensional vectors containing the *average*, *standard deviation*, and *skewness* of pixel values for each of the red, green, and blue channels, see [SO95]. An image is divided into five overlapping regions, each one represented by a 9-dimensional color feature vector. That results in a 45-dimensional vector as a descriptor of one image. The distance function used to compare two feature vectors is the well known Euclidean (L_2) distance. The second data set, called HV2, contains color histograms represented in 32-dimensions. This data set was obtained from the UCI Knowledge Discovery in Databases Archive ([Bay99]). The color histograms are extracted from the Corel image collection as follows: the HSV space is divided into 32 subspaces (32 colors: 8 ranges of hue and 4 ranges of saturation). The value in each dimension of such vector is the density of the corresponding color in the entire image. The distance function used to compare two feature vectors is the histogram intersection implemented as the *city-block* or *Manhattan* (L_1) distance.

4.2 Experimentation Setting and Obtained Results

Our implementation of the approximate similarity search algorithms was obtained by modifying the original code of the M-Tree search algorithms [CPZ97]. The similarity range queries were tested for a number of query radii, where the smallest and largest radii were chosen to retrieve approximately 1% and 17% of the data file, respectively. For each selected radius, range queries were executed for several values of proximity threshold x to see the change in search efficiency and effectiveness. In case of nearest neighbors queries, the number of neighbors k varied from 1 up to 50. We used the same values of proximity threshold as for the range queries.

In order to reduce statistical errors, each approximation level x was studied for 50 different query objects, not present in the data sets. Average values were computed for relative distance error ϵ, and improvement in efficiency IE. Obtained results are summarized in Figure 2. Notice that results are only shown for values of recall greater than 0.

The general observations can be summarized as follows. The best result, i.e. improvements of efficiency even of two orders of magnitude, are achieved when the query response set size is small (the smaller the better). The number of retrieved objects is explicitly specified for the nearest neighbors queries, but it is quite difficult to control it by specifying a range. Notice that the response sets for our range queries contained more than 100 objects, because 1% of 10 000 is 100. Consequently, the approach offers better performance for nearest neighbors queries with small k rather than the range queries.

The performance deteriorates with a growing number of nearest neighbors, and differences can mainly be observed between $k = 1$ and $k = 10$. However, the performance of queries for $k > 10$ becomes practically stable up to at least $k = 50$, so we do not show graphs for larger values of k.

The improvement of efficiency seems to be more significant for the 45 rather than the 32-dimensional vectors. We have also experimented with other data files and the general conclusion is that the method is mainly suitable when the precise similarity search tends to access most of data nodes in the supporting tree structure. Such situation often happens when the data partitioning results in highly overlapping regions, which is typical for high-dimensional vector spaces. Depending on the data file size, the improvements are typically registered in hundreds, which is not possible to achieve for low dimensional spaces where even the precise similarity search is already efficient.

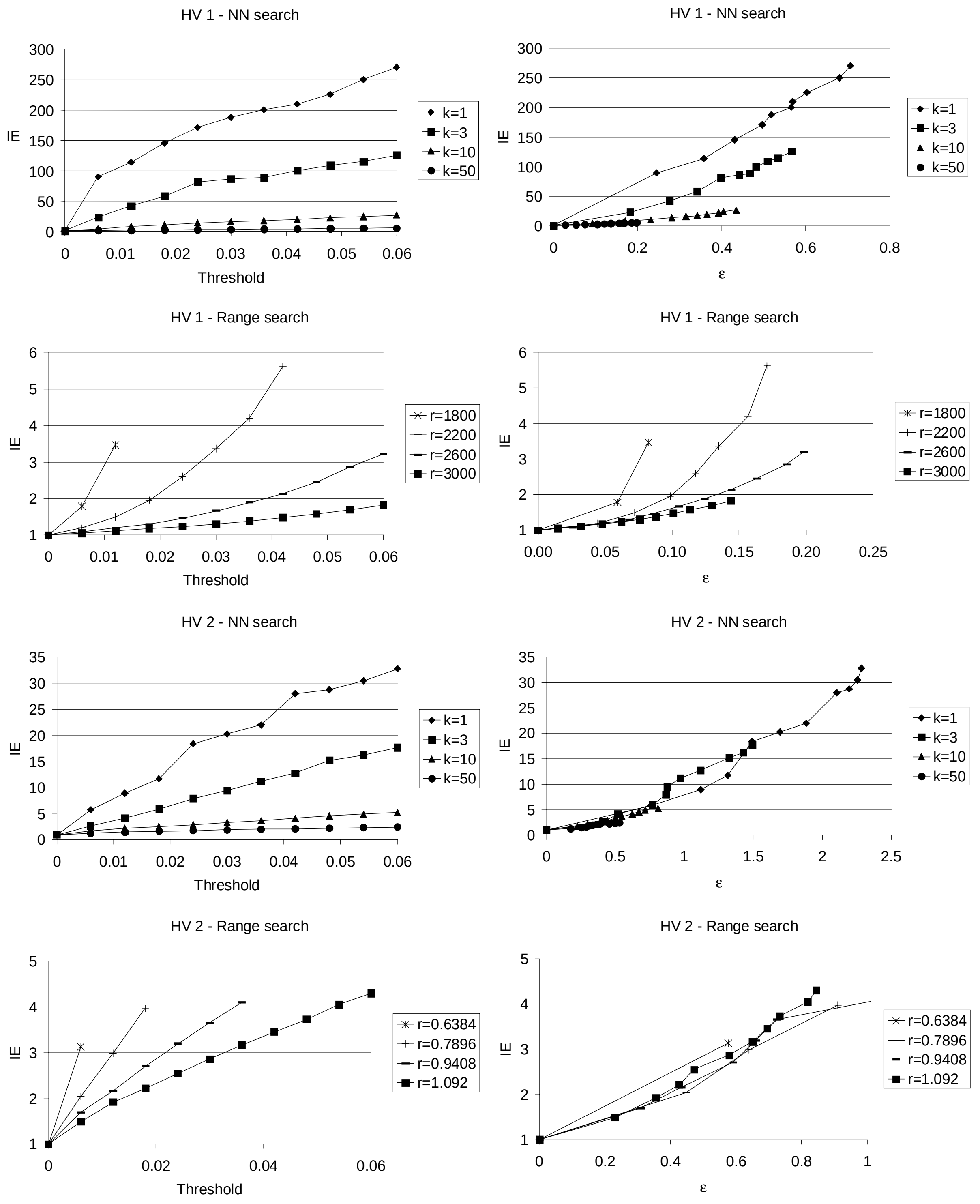

Fig. 2. Experimentation results

5 Conclusions

In this paper, we proposed and experimentally evaluated a new approach to the similarity retrieval with approximate results. We have considered generic metric spaces so our results are also valid for the important class of vector spaces, including high-dimensional vectors where the precise similarity search is limited by the *dimensionality curse* problem.

To evaluate a similarity query in tree organizations, typically many data nodes are accessed, even though not all of them finally appear to contain relevant data. In order to speedup the retrieval, we have proposed not to access nodes that have low probability of containing relevant data, that is when the proximity of data and query regions does not exceed specific threshold. We have specified a number of measures to properly assess the tradeoff between the achieved speedup and the quality of approximation. Experimental results on real-life data files are positive, and efficiency improvements of two orders of magnitude can be achieved for very precise approximations.

References

[ARS+00a] G. Amato, F. Rabitti, P. Savino and P. Zezula. Proximity Measure to Support Search through Distances. IEI-CNR Technical Report, [http://pc-erato2.iei.pi.cnr.it/amato/papers/proximityTR.pdf].

[ARS+00b] G. Amato, F. Rabitti, P. Savino and P. Zezula. Estimating Proximity of Metric Ball Regions for Multimedia Data Indexing. *ADVIS2000, First Biennial International Conference on Advances in Information Systems*, Izmir, Turkey, October 2000.

[AMN+98] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu. An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions. *Journal of the ACM*, 45(6):891-923, November 1998.

[Bay99] Bay, S. D. The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science.

[BGR+99] K.S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is Nearest Neighbor Meaningful? *ICDT'99*, pp. 217-235, Jerusalem, Israel, January 1999.

[BKK96] S. Berchtold, D.A. Keim, and H.P. Kriegel. The X-tree: An Index Structure for High-Dimensional Data. Proceedings of the *VLDB96*, Bombay, India, 1996.

[BO97] T. Bozkaya and Ozsoyoglu. Distance-Based Indexing for High-Dimensional Metric Spaces. *Proceedings of the 1997 ACM SIGMOD Conference*, Tucson, pp. 357-368, 1997.

[BO99] T. Bozkaya and Ozsoyoglu. Indexing Large Metric Spaces for Similarity Search Queries. *ACM TODS*, 24(3):361-404, 1999.

[CP00] P. Ciaccia, and M. Patella. PAC Nearest Neighbor Queries: Approximate and Controlled Search in High-Dimensional and Metric Spaces. Proceedings of the *16th International Conference on Data Engineering*, 28 February - 3 March, 2000, San Diego, California, USA. IEEE Computer Society 2000, pp. 244-255.

[CPZ97] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. *Proceedings of the 23rd VLDB Conference*, pp. 426-435, 1997.

[Gu84] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, pages 47–57, Boston, MA, June 1984.

[PAL99] S. Pramanik, S. Alexander and J. Li. An Efficient Searching Algorithm for Approximate Nearest Neighbor Queries in High Dimensions. *ICMCS 1999* IEEE International Conference on Multimedia Computing and Systems, 7-11 June, 1999, Folrence, Italy, Vol. 1 IEEE Computer Society.

[SO95] M. Stricker and M. Orengo. Similarity of Color Images. In: *Storage and Retrieval for Image and Video Databases III*, SPIE Proceedings 2420, 1995, pp. 381-392.

[TTS+00] C. Traina, A.J. Traina, B. Seeger, and C. Faloutsos. Slim-Trees: High Performance Metric Trees Minimizing Overlap Between Nodes. *Proceedings of the 7th EDBT International Conference*, pp. 51-65, March 2000, Konstanz, Germany.

[WSB98] R. Weber, H.-J. Schek, and S. Blott. A Quantitative Analysiss and Performance Study for Similarity Search Methods in High-Dimensional Spaces. *VLDB'98*, pp. 194-205, New York, NY, August 1998.

[ZSA+98] P. Zezula, P. Savino, G. Amato, and F. Rabitti. Approximate Similarity Retrieval with M-trees. *VLDB Journal*, 7(4):275-293, 1998.