Finding Aggregate Nearest Neighbor Efficiently without Indexing

Yanmin Luo[1,2]
[1] Dept. Computer Science, HuaQiao University
QuanZhou, FuJian, China, 362021
lym@hqu.edu.cn
[2] Dept. Computer Science, University of Tsukuba
Tennoudai 1-1-1, Tsukuba, Ibaraki 305, Japan
lym@dblab.is.tsukuba.ac.jp

Hanxiong Chen
Dept. Computer Science, University of Tsukuba
Tennoudai 1-1-1, Tsukuba, Ibaraki 305, Japan
chx@dblab.is.tsukuba.ac.jp

Kazutaka Furuse
Dept. Computer Science, University of Tsukuba
Tennoudai 1-1-1, Tsukuba, Ibaraki 305, Japan
furuse@dblab.is.tsukuba.ac.jp

Nobuo Ohbo
Dept. Computer Science, University of Tsukuba
Tennoudai 1-1-1, Tsukuba, Ibaraki 305, Japan
ohbo@dblab.is.tsukuba.ac.jp

ABSTRACT
Aggregate Nearest Neighbor Queries are much more complex than Nearest Neighbor queries, and pruning strategies are always utilized in ANN queries. Most of the pruning methods are based on the data index mechanisms, such as R-tree. But for the well-known curse of dimensionality, ANN search could be meaningless in high dimensional spaces. In this paper, we propose two non-index pruning strategies in ANN queries on metric space. Our methods utilize the r-NN query and projecting law, analyze the distributing of query points, find out the search region in data space, and get the result efficiently.

Categories and Subject Descriptors
H.2 [**Database Management**]; H3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms: Algorithms, Experimentation

Keywords
Spatial database, Aggregate Nearest Neighbor, Search Region

1. INTRODUCTION-MOTIVATION
Given the static source dataset $P=\{p_1,p_2,\ldots p_N\}$ and the query points set $Q=\{q_1,q_2, q_3 \ldots\ldots,q_n\}$, an ANN query retrieves the point in P which minimizes an aggregate distance function with respect to all the points in query set Q. The aggregate distance between a data point and query points set Q can be expressed by $adist(p,Q)= f(|pq_1|,|pq_2|,\ldots,|pq_n|)$, where $|pq_i|$ is the Euclidean distance between point p and q_i. Different function f gives ANN query different meaning. ANN query has received considerable attention the last few years and it has become more and more important in spatial database [1,2,3]. In order to compute as few distances as possible, pruning strategies are always used to optimize the query processing in ANN queries, for which many efficient indexing structures have been proposed such as R-tree. But for the well-known curse of dimensionality, the traditional indexing methods are reasonably well solved just for low dimensional applications. Many studies have shown that traditional indexing methods fail in high dimensional space. Thus NN search and ANN search would be meaningless in high dimensional spaces.

In this paper, we propose two non-indexing pruning strategies for ANN query processing which we call them vp-ANN algorithm and projection-based algorithm. We assume that all query points can fit in main memory and only consider the sum function. For the following discussion, we consider 2-Dimendional point datasets. But the proposed techniques are applicable to higher dimension

Infoscale 2007, June 6-8, 2007, Suzhou, China

2. PRUNING WITHOUT INDEX
As for the sum function in ANN query, $adist(p,Q)=sum(|pq_1|, |pq_2|,\ldots,|pq_n|)$, the best ANN point should make the value of distance of $|pq_j|_{j=1..n}$ be as small as possible. It is clearly that the best ANN result would be lie in the region in which the query points distribute concentrically. If the data points in dataset P distribute uniformly in data space. We can say that the result of ANN query (we call it the best ANN point) should be inside the region which the MBR(Minimum Bounding Rectangle) of the query points set Q covers in data space. In our methods, pruning means we find a search region in data space and get the best ANN point in this region, instead of searching in full data space. Not using the indexing mechanisms, our pruning methods analyze the distributing of query points by different ways, and mark out the search region. The most important technique of our methods is how to find the search region which is equal to or covers the MBR of query points set.

2.1 Vp-ANN Algorithm
The ideal best ANN point p would be the point which lets every $|pq_j|_{j=1..n}$ be minimal in $\{|p_iq_j|_{i=1..N}\}$. This is to say this point would be the Nearest Neighbor of every query point q_j(j=1..n) synchronously. For the data points of P distribute uniformly and there must be many points in P, we can say this ideal best ANN

point p would not exist indeed. But enlightened by this, we know the best ANN point p would be as near as possible to each query point in Q. Given a point vp and radius r, we describe the r-neighbor region of vp by C(vp,r)={p|p$\in$ P, |p vp|<=r}. It can be concluded that for every point q_i the best ANN point should be in its r-neighbor region. On that we give out Lemma1.

Lemma1. Let $R_i=C(q_i,r_i)$ be the r_i-neighbour region of q_i, and $R=\cup_{i=1..n} R_i$ be the union area of R_i. If each r_i we choose can guarantee that the intersection of $\cap_{i=1..n} R_i$ is not empty, the best ANN point must be in R.

The R in Lemma1 is the search region of our vp-ANN algorithm. Now the most important thing for us is to consider each r_i carefully. For simplifying the process, we choose a point vp in dataset P, and select the Euclidean distance between vp and q_i to be r_i. The focus converts to how to select the pivot vp. We know the aggregate centroid of Q is the best choice of vp, because as for sum function this point is the one in data space that minimizes the value of aggregate distance. We can say that the aggregate nearest neighbor is a point of P which 'near' the aggregate centroid of Q. However it is difficult and time-consuming to calculate the aggregate centroid. Thus we use the geometric centroid to substitute it. The geometric centroid: q(x,y) of Q can be calculated by Eq.1 and Eq.2.

$$x = (1/n)\cdot\sum_{i=1..n} x_i . \quad (1)$$

$$y = (1/n)\cdot\sum_{i=1..n} y_i. \quad (2)$$

Although the geometric centroid of query points set Q is the best choice of vp, it is not certain a point of the dataset P. if we choose it as vp, it is possible that the search region R would be a empty set. For the reason we find the nearest neighbor of the geometric centriod in P (Assuming this point is p_k), and choose it(p_k) as the vp point instead of the geometric centriod. On this vp, there is at least one point p_k in the search region R. For the search region always covers the MBR of query points set, vp-ANN algorithm is strong and always can get the best ANN query point.

2.2 Projection-based Algorithm

The heart of the projection-based method is to project the query points into a carefully selected 'line'. The projecting result can reflect the distributing of query points. By the distributing of the projecting points in the line we can determine the search region in which the query points distribute concentrated, and prune other region. The search region of projection-based algorithm is the intersection of two candidate regions. The candidate region is a cirque between the two circles. Both of the candidate regions contain all the query points.

In order to calculate the candidate region, projection-based method chooses two points p_a and p_b in data set P, and considers the 'line' that passes through p_a and p_b in data space, then projects the query points into the line. The distances between pivot p_a and the projecting points of the query points on the line can be calculated by the Eq.3, which can be deduced by cosine law. In the equation $D(p_a,q)$ is the Euclidean distance between p_a and q. In order to let our algorithm be more strong, the points p_a and p_b we select should maximize the distance $D(p_a,p_b)$. This means p_a and p_b must be in the brim of data space. We choose this two pivots by special method which requires just O(N) distance computations.

$$Proj(p_a,q)=(D^2(p_a,q)+D^2(p_a,p_b)-D^2(p_b,q))/(2D(p_a,p_b)) \quad (3)$$

After calculating all the value of $proj(p_a,q_i)(i=1..n)$, we find out the query point q_{max} and q_{min} , which the value of $proj(p_a,q_{max})$ and $proj(p_a,q_{min})$ are the maximum and minimum respectively in $proj(p_a,q_i)(i=1..n)$. Let q_{max}' be the projecting point of q_{max} on the line and q_{min}' be the projecting point of q_{min} on the line. For insuring that the candidate region would cover all the query points we calculate the radii of the candidate region by Eq.4 and Eq.5.

$$r_{max} = proj(p_a,q_{max})+D(q_{max},q_{max}'). \quad (4)$$

$$r_{min} = proj(p_a,q_{min}) - D(q_{min},q_{min}') \quad (5)$$

In Eq.5 if $r_{min} < 0$, we let $r_{min} = 0$. Taking p_a as the centre of a circle we draw two circles with the radii r_{max} and r_{min} respectively, the cirque between the two circles is the candidate region of projection-based method. We can express the candidate region by $A(p_a,r_{min},r_{max}) =\{p|p\in P, r_{min}<=|p_ap|<=r_{max}\}$.

Because the candidate region covers all the query points, it is certainly to cover the MBR of all the query points. Therefore this region is also a strong one and would include the best ANN point. However the range of the candidate region is still large, and it may include almost of the points in dataset. If we take this candidate region as search region, we just can prune a few points in data space. For get the search region which is much smaller than the candidate region, our strategy is: firstly, we select two points p_{a1} and p_{b1}, and get the frist candidate region: $A(p_{a1},r_{min1},r_{max1})$. Then we select the other two points p_{a2} and p_{b2} which are different from p_{a1} and p_{b1}, and get the second candidate region $A(p_{a2},r_{min2},r_{max2})$. Because the two region have different circle centres p_{a1} and p_{a2}, and we project just the same dataset P into the two different lines, there must be an intersecting region between $A(p_{a1},r_{min1},r_{max1})$ and $A(p_{a2},r_{min2},r_{max2})$. This intersecting region would much smaller than each one of the two candidate regions, and We symbolize this region by $S= A(p_{a1},r_{min1},r_{max1})\cap A(p_{a2},r_{min2},r_{max2})$. We select S as the search region of projection-based algorithm. Because both the two candidate regions cover the MBR of the query points, this search region is a "satisfied" one.

3. EXPERIMENTS

We use both real datasets and synthetic datasets in our experiments. The result shows that the value of n(point number of Q) and the area of MBR of Q are the important factors of our algorithms. And as for CPU cost, the projection-based algorithm is better than vp-ANN algorithm. Our methods perform well in high dimension space.

4. REFERENCES

[1] Dimitris Papadias, Yufei Tao, Kyriakos Mouratidis, Chun Kit Hui: Aggregate nearest neighbor queries in spatial databases. ACM Trans. Database Syst. 30(2): 529-576 (2005)

[2] Hongga Li, Hua Lu, Bo Huang, Zhiyong Huang: Two ellipse-based pruning methods for group nearest neighbor queries. GIS 2005: 192-199

[3] Man Lung Yiu, Nikos Mamoulis, Dimitris Papadias: Aggregate Nearest Neighbor Queries in Road Networks. IEEE Trans. Knowl. Data Eng. 17(6): 820-833 (2005)