ANNの効率的なフィルタリング

王銘新 陳漢雄 古瀬一隆 大保信夫
筑波大学システム情報工学研究科CS専攻

1. はじめに

ANN検索（集団的な近傍検索）はNN検索から発展し、複数のquery（問い合わせ）を対象として、定められた距離関数に基づき、近傍検索を行う検索である。NN検索と同様、DM、IR、特に、GIS（地理情報システム）で不可欠なものとなっている。[3]

一般的には、NN検索には距離計算が常に伴っている。この距離は近似度を表わしている。データベースが大きいほど、距離比較のコストが高くなる。

ANNにqueryポイントが複数存在するため、距離計算はさらに重くなり、検索システムに大きな負担となっている。

この負担を減らすために、新たなフィルタリング（絞り込み）手法を提案する。近似度の低いデータを削減し、近似度の高いデータだけに集中すれば、計算コストは大幅に押さえられる。精確かつ高速な検索が望まれる。

ANNの距離関数は単調関数である。本研究では、SUM関数を対象としている。

2. 関連研究

BF-NN(best first nearest neighbor)はR-treeを用いた単一queryの近傍検索である。

文献[1]より提案された手法である。

3. 提案手法

座標空間のデータベースと距離関数はSUMだけを前提条件として限定する。

図1のように、まず各queryポイントの重心に一番近いデータポイント（p3）を求め、各query点を円心にこの重心との距離を半径とし、円を作る。これらの円にカバーする領域をD_qとする。D_qの外側にあるデータポイントp2とp4はp3より遠いので、排除できる。一方、D_qの内部にあるp1については実際の距離を計算しなければp3よりよいかどうか分からない。言いかえると、問い合わせの答えはD_qから調べればよい。

Efficient Filtering for Aggregate Nearest Neighbor Search
MingXin Wang, Hanxiong Chen, Kazutaka Furuse, Nobuo Ohbo
Graduate School of SIE, University of Tsukuba

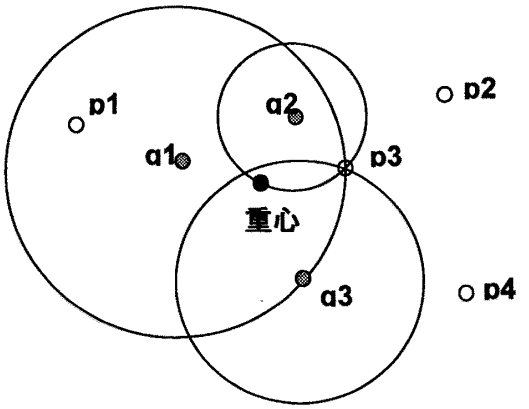

図1 理想的な探索領域

3.1. フィルタリング

まず、本文で使用する記号を説明する。

D：ベクトルデータ集合

N：データポイントの数、即ちN=｜D｜

p_a, p_b：Dの中最も互いに離れている2つのデータポイント

Q：queryポイントq_iの集合

M：queryポイントの選択範囲

G：queryポイント集合Qの重心

vp：Gの最近傍になるデータポイント

dist (p_i, q_j)：データポイントp_iとqueryポイントq_jの距離

amax、amin：p_aによるフィルタリングの上界と下界

bmax、bmin：p_bによるフィルタリングの上界と下界

まず、索引を作成する。

p_a、p_bを探し出す。全部の$p_i \in D$に対してdist(p_a, p_i)とdist(p_b, p_i)を計算し、格納する。

この索引構造を用いてフィルタリングを次のように行う。

1. Gを計算する。
2. BF-NNでGのNNとなるvpを探す。
3. amax=Max(dist(p_a, q_i)+dist(vp, q_i))、
   amin=Min(dist(p_a, q_i)-dist(vp, q_i))、
   bmax=Max(dist(p_b, q_i)+dist(vp, q_i))、
   bmin=Max(dist(p_b, q_i)-dist(vp, q_i))を計算する。
4. For all p_k in D if (dist(p_a, p_k)>amax) or (dist(p_a, p_k)<amin) or (dist(p_b, p_k)>bmax) or (dist(p_b, p_k)<bmin) then p_kを排除する。
5. 残るデータポイントの実距離を計算し、答えを出す。

以下で例を用いて上記アルゴリズムを説明する。

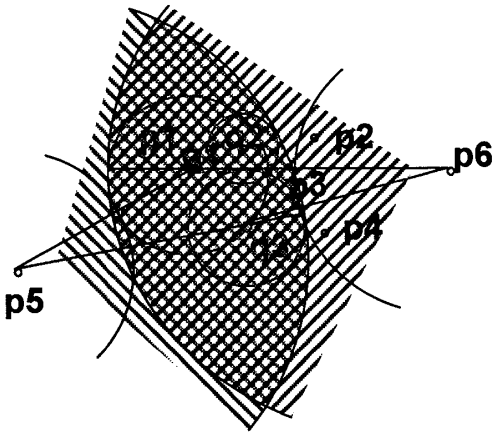

図２　フィルタリング範囲

前処理で、p_a=p5、p_b=p6 が分かった。

問い合わせとして、q1、q2、q3 が与えられたとする。まず、それらの重心を計算する。重心の計算に簡単な式 $G=(1/n)\Sigma_{i=1..n}q_i$ を用いる。次に、vp=p3 はすぐ見つかった。よって、amax、amin、bmax、bmin は計算できる。

ステップ４は肝心なフィルタリングである。このフィルタリングの結果は図２のチェック模様（フィルタリングの選択範囲）にあるデータポイント p1 と p3 しか残らない、つまり、p2、p4、p5 と p6 は排除される。最後に、p1、p3 との SUM 距離を計算し、答え p3 は得られる。

計算コストは主に三つ、vp の計算、dist(q_i, vp) およびフィルタリングした後に残るデータポイントの距離計算にかかる。明白に、三番目のほうは一番コストが高い。この計算コストを減らすため、高いフィルタリング効果による多くのデータの事前排除がのぞましい。

データベースが定まっている以上、残るデータポイントの数は図２のチェック模様の面積で決める。query ポイントがどんなに増えても、フィルタリングの選択範囲が広くならない限り、計算量は増加しない。

3.2. 実験と考察

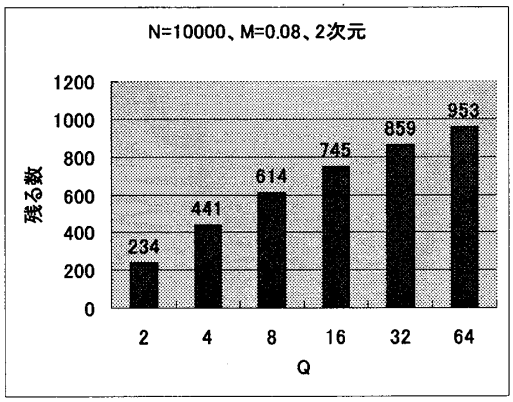

図３　query 数の増加実験

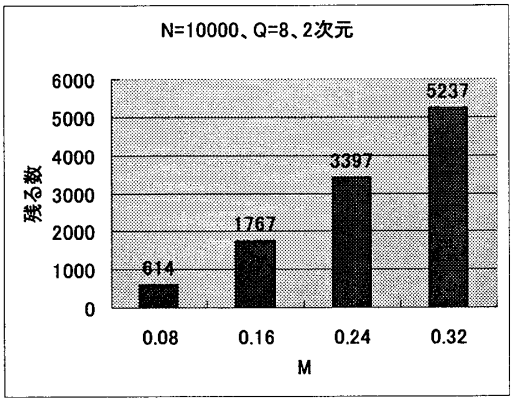

図４　query 範囲の増加実験

乱数で一万個のデータを生成し、実験を行った。図３と図４は全部一万回の実験を繰り返し、平均値を取った結果である。

図３に、Q（query の個数）の倍増につき、残るデータポイントの数は収束することが分かる。フィルタリングの効果はほぼ 90%以上なることが示された。

一方、図４のように、M（query の範囲）の増大に伴い、残るデータポイントの数はシャープに上がる。フィルタリングの効果はだんだん得られなくなる。

これも現存の ANN アルゴリズムの共通的な問題点であり、いまだに解決できずになっている。[2]

4. おわりに

ANN の効率的なフィルタリング手法を提案した。

MQM との計算時間の比較実験は今後の課題とする。さらに、もう一度基本から考え直し、query 範囲の難題を解決したい。

謝辞

本研究の一部は科学研究費補助金特定領域研究（＃19024006）による。

参考文献

[1] Dimitris Papadias, Yufei Tao, Kyriakos Mouratidis, and Chun Kit Hui：Aggregate Nearest Neighbor Queries in Spatial Databases, ACM Transactions on Database Systems, Vol.30, No.2, June 2005, Pages 529-576

[2] Yanmin Luo, Hanxiong Chen, Kazutaka Furuse, and Nobuo Ohbo：Efficient Methods in Finding Aggregate Nearest Neighbor by Projection-based Filtering, ICCSA, vol.4707, 2007

[3] Man Lung Yiu, Nikos Mamoulis, and Dimitris Papadias：Aggregate Nearest Neighbor Queries in Road Networks, IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.6, June 2005