

逆遠方検索とその効率的な検索方法に関する研究

市川 敦啓[†] 陳 漢雄^{††} 古瀬 一隆^{††}

[†] 筑波大学大学院システム情報工学研究科 〒305-8573 茨城県つくば市天王台 1-1-1

^{††} 筑波大学システム情報系 〒305-8573 茨城県つくば市天王台 1-1-1

E-mail: [†]tichikawa@dblab.is.tsukuba.ac.jp, ^{††}{chx,furuse}@cs.tsukuba.ac.jp

あらまし データ工学で研究されている分野、類似検索のバリエーションの一つに、逆最遠方検索がある。これはデータセット O 及びクエリ q が与えられたとき、 O の中から q を最遠方とするような全てのオブジェクトを求めるという検索手法である。逆最遠方検索は施設配置に関する最適化問題などに非常に有用であるが、最遠方のみを考えるため、尺度として問題がある。そこで本研究では、一般的な逆遠方検索を提案し、新たに2つの検索アルゴリズム (RFarN, RkFN) を考案する。さらにその効率的な検索手法を提案し実験を行い、それらの有用性を検証する。

キーワード 問合せ処理, 距離索引, 逆遠方検索

1. はじめに

近年、著しい増加の一途をたどるデータに対する類似検索が注目され、科学研究、開発への応用がなされている。例として、パターン認識 [1], [2] や、画像検索 [3]、時系列マッチング [4] などが挙げられる。これらの応用に伴い、類似検索に関する研究は活発にされている。

各分野において、しばしばデータはベクトルとして表される。例えば、地図データの座標や、画像データのピクセル・色情報などを多次元の浮動小数点型で表す。このようなデータの数値を座標系に置いて、ベクトル間の距離を計算することで、データの類似度を求めることができる。一般的に、距離の小さいデータほど類似度が高いといえる。

これまで、様々な距離検索が提案されており、それぞれの手法に意義がある。ここで、それらの検索手法の紹介とともに、実世界での実用例を挙げる。

近傍検索 (Nearest Neighbors, NN Search) は類似度が高いデータの検索であり、例えば GPS 情報を用いて現在地から近い店を検索するといった例がある。これまで、類似度の高い一定の数のデータを検索する k 近傍 (k -NN) や、一定の類似度をもつデータを探索範囲検索 (Range Query) といった手法が研究されてきた。

遠方検索 (Furthest Neighbors, FN Search) は近傍検索とは逆に類似度の低いデータの検索である。

逆近傍検索 (Reverse NN, RNN Search) は相手から見て自分が近傍となるようなデータの検索であり、近傍検索とは異なる。例えばスーパーなど店舗を配置するときは、そのスーパーが他の店より近いと思う客が多いほどよい。すなわち多くの客にとってスーパーが近傍になるような場所を検索したい。このような場合に逆近傍検索は有用である。

逆遠方検索 (Reverse FN, RFN Search) は相手から見て自分が遠方となるようなデータの検索であり、同じくスーパーを配置するときには、なるべく競合を避けるために他のスーパーから見て遠くに配置したいといった場合に用いられる。

他の例では、Twitter において、あまり自分に感心がなく、たとえフォローしてもあまりフォロー返しは期待できないユーザを見つけることができる。一般ユーザにとってはあまり意義がないように思えるが、例えば政治家にとっては自分の支持が薄い層、すなわち特に力を入れて宣伝すべきユーザ層を見つけることができるなど、ビジネスでの応用の可能性をもつ検索であるといえる。

ビジネス業務においては、顧客の要求を数値化しベクトルで表し、同じく商品データもベクトルにすることで、要求ベクトルとの距離を計算できる。その距離が小さいほど、商品が顧客の要求に相応しいことがわかる。顧客にとっては自分の要求に合った商品を探したいため、近傍検索が必要となる。一方で商品を扱うメーカー側は製品の”顧客受け”を知りたい状況にあり、特に”顧客受け”がよくない製品を探すことは改善作業の役に立つ。ゆえにメーカーにとっては遠方検索も重要であるといえる。

これらの検索手法のうち NN, FN, RNN に関してはこれまで盛んに研究され様々な効率的なアルゴリズムが提案されてきたのに対し、RFN は歴史が浅く、2009 年に初めて逆最遠方検索が提案された [5]。

逆最遠方検索はその名の通り、クエリを最も離れた点とするデータの集合を求める検索である。しかし、最も離れた点という条件が厳しく、実世界での応用には不都合な場合がある。例えばデータが図 1 のような場合は、 q を最遠方とする点は存在せず解は見つからない。 o_8 のようなデータの存在の影響で、本来見つけたい o_3 や o_4 のようなデータも発見できなくなってしまう。このような少数の外れ値データの存在で、有用性が大きく損なわれてしまうという問題点がある。

逆遠方検索が有用な例には、施設の配置問題や商品の人気傾向調査といった、度合いが関わるものが多い。本質的にそれらは、クエリと十分離れているデータを探したい。しかし、逆最遠方検索では実際にクエリとデータが十分遠い位置にあっても解にならず、意図した検索結果にならない場合がある。このことから逆最遠方検索は時として尺度として問題があるといえる。

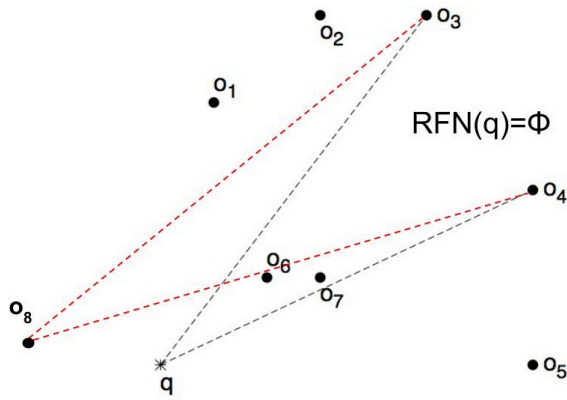


図 1 逆最遠方検索

そこで本研究では、逆最遠方検索において解の条件を緩和するという着想で、新たな 2 つの問い合わせ問題を提案し、アルゴリズムを考案する。さらに、提案した検索問題に対する効率的な検索手法を考える。

本稿では、続く第 2 章で逆最遠方検索の効率的なアルゴリズムを紹介し、第 3 章で新たな検索問題を提案し、その効率的な検索アルゴリズムを考える。第 4 章で実験を行い提案したアルゴリズムの有用性を示し、第 5 章で結論を述べる。

2. 関連研究

類似検索は一般的に計算コストが高い。それゆえにこれまで効率化を目的とした研究が多く行われてきた。また、逆最遠方検索は、近傍検索の研究で提案された手法や効率化の技術を多く受け継いだ。本章では、これまで研究された逆最遠方検索の手法を紹介する。

近傍検索には、R-tree [6] を用いる深さ優先探索法 [7], [8] と、最良優先探索法 [9] が早期に提案された。近年には距離に基づくハッシュ法を用いた近似最近傍検索 [10] や、LSB-tree によって高次元データに対応した最近傍検索手法 [11] が提案された。

逆最遠方検索には、2009 年に初めて RFN 問題を定義した Yao らは Convex Hull Furthest Cell (CHFC) アルゴリズムを提案した [5]。これは R-tree による空間分割索引法や、凸包定理・ボロノイセルを用いたフィルタリングを行うことで効率化が実現した。これに対して、Liu らは 2010 年に距離索引と凸包定理・三角不等式によるフィルタリングを用いた PIV アルゴリズムを提案し、更なる効率化を実現した [12]。

ここで、これらのアルゴリズムに共通して根拠となる理論、凸包定理について説明する。図 2 のように、データセットの外周を線分でつなぎ、全オブジェクトを包囲するような区域 (凸包) をつくる。このとき、クエリの逆最遠方解が空ではないための必要十分条件は、クエリが凸包の外側あるいは境界線にあることである。この定理の厳密な証明は [5] で示されている。

Liu らの PIV アルゴリズムでは、R-tree は使わず、代わりに事前にデータの距離情報を索引表 (Metric-Index) に保存する。凸包と索引表の作成は前処理で一回のみ実行する。検索クエリが与えられると索引表に基づいた走査を繰り返す。検索時は凸

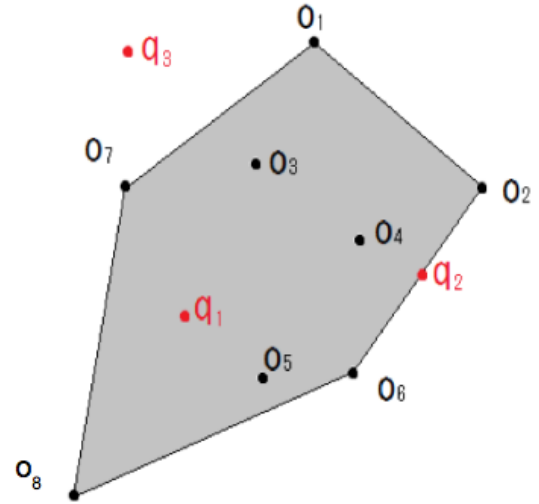


図 2 凸包定理により、 q_1 の逆最遠方解は存在せず、 q_2, q_3 の逆最遠方解は存在する。

包定理と三角不等式を利用したフィルタリングを行う。

3. 提案手法

これまでの逆最遠方検索に関する研究は、逆最遠方を探すことだけに焦点をあてて効率化が検討されてきた。しかし、先に述べたとおり、逆最遠方だけでは条件が厳しく実世界での応用が極めて限定されてしまう。そこで本研究では、一般的な逆最遠方問題を 2 つ提案し、その検索アルゴリズムを考案した。1 つ目は逆最遠方から距離 d の範囲を許容する「逆最遠方範囲検索」(RFarN)、そして 2 つ目は k 番目まで遠くにクエリを持つデータを全て許容する「 k 逆最遠方検索」(RkFN) である。これらの概要を説明し、検索アルゴリズムを提案し効率化を検討する。

3.1 逆最遠方範囲検索 (RFarN)

クエリ q に対し、あるデータオブジェクト o から o の最遠方 $FN(o)$ までの距離と、 o から q までの距離の差がパラメータ d 以内の場合 o を解とする。

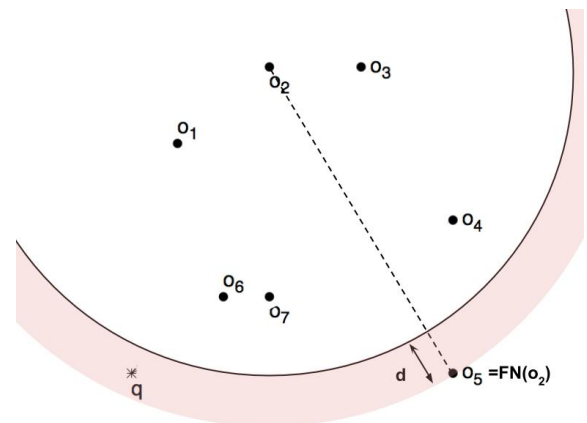


図 3 例えば、 o_2 は q の逆最遠方ではないが、許容する範囲を d とすると、 o_2 からその最遠方 o_5 までの距離と、 o_2 から q までの距離の差が d 以下のため、 o_2 も解に含まれ、 $RFarN(q, d)$ の解は $\{o_2, o_3, o_4\}$ 。

以下に、RFarN の定義を示す。式中の $dist(a, b)$ は点 a, b 間の距離を表し、許容する範囲を d とする。またオブジェクト o の集合を O とする。

[定義] (RFarN)

$$RFarN(q, d) = \{o | o \in O \wedge dist(o, q) \geq dist(o, FN(o)) - d\}$$

ただし $FN(o)$ は o の最遠方である

RFarN 検索では、従来の逆最遠方検索よりも柔軟な逆遠方検索が可能と考えられる。具体的な例では、施設配置において競合を「なるべく」避けたい場合に、他店から十分離れているものの逆最遠方検索では解にならないような場所を見つけることができる。しかし、Algorithm 1 で示すように RFarN 検索のナイーブなアルゴリズムの計算コストは、データセットのサイズを n とすると $O(n^2)$ と処理効率が悪い。そこで本研究では RFarN を効率的に検索するアルゴリズムを提案する。

Algorithm 1 RFarN(Dataset O , Query q , Range d)

```

1:  $ans \leftarrow \phi$ 
2: for  $o \in O$  do
3:    $\delta = 0$ 
4:   for  $v \in O$  do
5:     if  $dist(o, v) > \delta$  then
6:        $\delta \leftarrow dist(o, v)$ 
7:     end if
8:   end for
9:   if  $dist(o, q) \geq \delta - d$  then
10:     $ans \leftarrow ans \cup \{o\}$ 
11:   end if
12: end for
13: return  $ans$ 

```

3.2 RFarN の効率的な検索アルゴリズム

RFarN の効率化にあたり、既存の逆最遠方検索のアルゴリズムの利用を検討する。先に述べたとおり、逆最遠方検索には Yao らの CHFC アルゴリズムや Liu らの PIV アルゴリズムが存在している。このうち CHFC アルゴリズムで用いられた空間分割索引法は、RFarN では広範囲が解の候補となるため適していない。一方 PIV アルゴリズムでは距離索引とともに、凸包定理と三角不等式によるフィルタリングを行う。このうち、RFarN では範囲 d を許容するため凸包定理は利用できないが、距離索引と三角不等式によるフィルタリングは利用できると考えられる。

RFarN に PIV アルゴリズムを適応した場合の処理を説明する。データセットを O 、許容する範囲を d とする。前処理であらかじめそれぞれのデータ間の距離を計算し距離索引に保存する。同じく前処理で O の凸包を作成し、頂点をそれぞれ p とする。クエリ q が与えられた時、オブジェクト $o \in O$ が q の RFarN 解になるかを判定する。しかし、実際に $dist(q, o)$ を計算し比較せずとも o が解となるかを三角不等式フィルタリングによって判定できる場合がある。そのような場合を補題 1, 2 として以下に示す。なお以下の式において、 $dist(p, o), dist(o, FN(o))$ は

前処理であらかじめ計算され距離索引に保存されている。

[補題 1] o が解であるための必要条件

$$dist(p, o) + dist(p, q) < dist(o, FN(o)) - d \quad \text{ならば}$$

$$o \text{ は } q \text{ の RFarN ではない}$$

[補題 1 の証明] 補題 1 の条件と三角不等式により、以下の不等式が成り立つ。

$$dist(q, o) \leq dist(p, o) + dist(p, q)$$

$$< dist(o, FN(o)) - d$$

すなわち、 o によって q は最遠方の範囲 d よりも近くにあり、定義より解ではないことがわかる。(図 4)

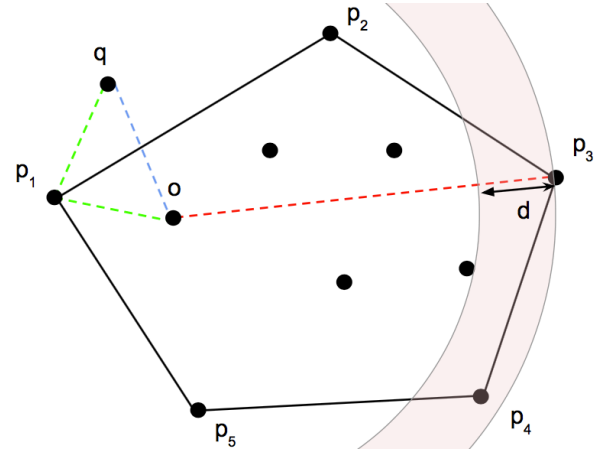


図 4 補題 1 について、三角不等式により $dist(q, o) \leq dist(p_1, o) + dist(p_1, q)$ が成り立つため、 $dist(p_1, o) + dist(p_1, q) < dist(o, p_3) - d$ ならば $dist(q, o) < dist(o, p_3) - d$ となり、 o は RFarN 解から排除される。

[補題 2] o が解であるための十分条件

$$|dist(q, p) - dist(o, p)| \geq dist(o, FN(o)) \quad \text{ならば}$$

$$\text{よって } o \text{ は } q \text{ の RFarN である}$$

[補題 2 の証明] 補題 2 の条件と三角不等式により、以下の不等式が成り立つ。

$$dist(q, o) \geq |dist(q, p) - dist(o, p)|$$

$$\geq |dist(q, p) - dist(o, p)| - d$$

$$\geq dist(o, FN(o)) - d$$

すなわち、 o によって q は最遠方の範囲 d よりも遠くにあり、定義より解であることが断定できる。

ナイーブなアルゴリズムでは、オブジェクト一つ一つに関し、実際にクエリとの距離を計算し、他のオブジェクトとの距離と比較する必要があるのに対し、このアルゴリズムではオブジェクトから各凸包頂点までの距離を比較するだけで、そのオブジェクトが RFarN の解であるかどうかを判定することができる。

Algorithm 2 RFarN(Dataset O , Pivot S_{piv} , Query q , Range d)

```
1:  $cand \leftarrow \phi$ ,  $ans \leftarrow \phi$ 
2: Compute each distance pair  $dist(q, p), p \in S_{piv}$ 
3: for  $o \in O$  do
4:   if  $\min_{p_i \in S_{piv}} \{dist(o, p_i) + dist(p_i, q)\} < dist(o, FN(o)) - d$ 
     then
5:     Discard  $o$  //補題 1
6:   else
7:     if  $\max_{p_i \in S_{piv}} \{|dist(o, p_i) - dist(p_i, q)|\} > dist(o, FN(o))$ 
       then
8:        $ans \leftarrow ans \cup \{o\}$  //補題 2
9:     else
10:       $cand \leftarrow cand \cup \{o\}$  //実距離計算が必要
11:    end if
12:  end if
13: end for
14: for  $o \in cand$  do
15:   Compute real distance  $dist(o, q)$ 
16:   if  $dist(o, q) < \max_{p_i \in S_{piv}} \{dist(o, p_i)\} - d$  then
17:     Discard  $o$ 
18:   else
19:      $ans \leftarrow ans \cup \{o\}$ 
20:   end if
21: end for
22: return  $ans$ 
```

アルゴリズムを Algorithm 2 に示す。Pivot は凸包の頂点を表し、その数を m とする。4,5 行目において補題 1、6-8 行目において補題 2 のフィルタリングを行う。フィルタリングが出来なかったオブジェクトに関しては 14 行目からクエリとの実距離の計算をすることになる。結果として計算コストは 3, 4, 7 行目から $O(nm)$ となり、ナイーブなアルゴリズムと比べ大幅な効率化が期待できる。

3.3 k 逆最遠方検索 (RkFN)

RFarN が逆最遠方からの距離で許容するのに対し、RkFN は逆最遠方からの数を許容する。あるデータオブジェクト o がクエリ q の逆最遠方でなくても、 o から見て遠い k 個の点に q が含まれる場合も o を解とする。以下に定義を示す。

[定義] (RkFN)

$$RkFN(q, k) = \{o | o \in O \wedge dist(o, q) \geq \min_{v \in kFN(o)} dist(o, v)\}$$

ただし $kFN(o)$ は o から遠い k 個のデータである

RkFN 検索も逆最遠方に準じた点を検索できると考えられる。Algorithm 3 に示すように、ナイーブなアルゴリズムの計算コストは RFarN と同様 $O(n^2)$ である。しかし、効率化に関しては、逆最遠方検索で用いられた凸包定理や三角不等式によるフィルタリングが適応できなくなるため、アルゴリズムを利用することができない。

3.4 RkFN の効率的な検索アルゴリズム

そこで効率化のために考えられるのは、逆最遠方解のフィル

Algorithm 3 RkFN ナイーブアルゴリズム (Dataset O , Query q , Number of Answer k)

```
1:  $ans \leftarrow \phi$ 
2: for  $o \in O$  do
3:    $count = 0$ 
4:   for  $v \in O$  do
5:     if  $dist(o, q) < dist(o, v)$  then
6:        $count \leftarrow count + 1$ 
7:     end if
8:   end for
9:   if  $count < k$  then
10:     $ans \leftarrow ans \cup \{o\}$ 
11:   end if
12: end for
13: return  $ans$ 
```

タリングである。RkFN は逆最遠方検索の条件を緩和した検索であるため、当然逆最遠方も解に含まれる。そこで、逆最遠方検索の解の部分を [12] の高速なアルゴリズムでフィルタリングし、残る点だけをナイーブなアルゴリズムで計算することで、全ての点をナイーブなアルゴリズムで計算するより高速化が期待される。

4. 実験

本稿の提案手法の有用性を示すため、速度・フィルタリング率の観点から検証実験を行った。本実験の環境は以下の通りである。

- CPU : Intel®COREi5 650 3.20GHz
- メモリ : 4GB
- 言語 : C++

実験データには合成データと実データを用いた。合成データはランダムクラスタ分布 (rc) と一様分布 (un) の 2 種類を、実データはアメリカの道路ネットワーク図 (map) を利用した。

これらのデータセットは 2 次元ベクトルで表される点の集合であり、値域はそれぞれ $[0, 120000]$ とする。

4.1 RFarN

3.2 節の提案手法による RFarN の効率化の検証のため、Algorithm 2 に示される提案手法と、Algorithm 1 に示される RFarN のナイーブなアルゴリズムのプログラムを実装し、それぞれのデータセットにおいて検索を行い、計算コストと検索結果の評価を行った。許容する範囲を $d = 10000$ としたときの結果とその計算コストをそれぞれ図 5, 6 に示す。

図 5 は map のデータセットを用いて RFarN と逆最遠方検索 (RFN) の解を比較した結果である。RFarN では、RFN の解以外にもクエリが十分遠くと判断できる点も解となっていることがわかる。 d を変化させることによって尺度を調整でき、とくに $d = 0$ のときは RFN そのものになるため、RFN に比べ柔軟な検索手法であるといえる。

図 6 の横軸はデータセットのサイズ、縦軸はクエリ 1 つあたりの平均計算コストを示す。データセットの数が大きくなるにつれ、ナイーブなアルゴリズムは著しく計算コストが増大

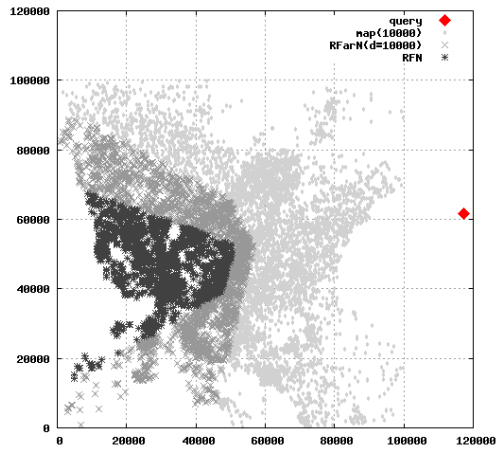


図 5 RFN と RFarN の比較 (d=10000)

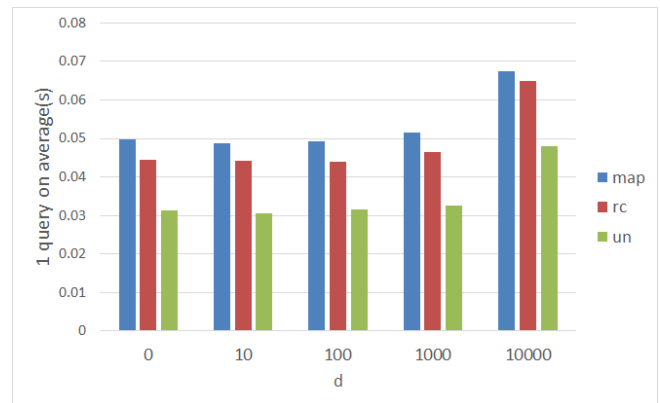


図 7 d に応じた計算コスト

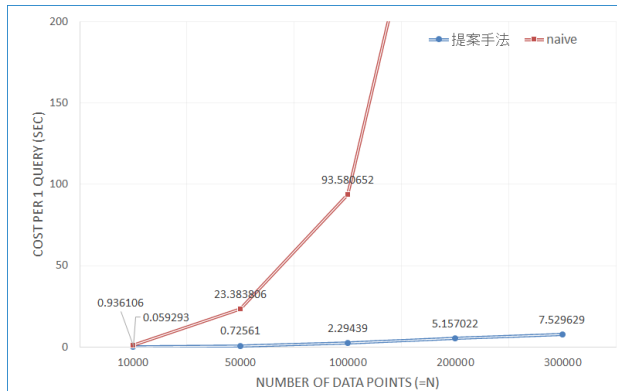


図 6 RFarN 提案手法 VS ナイーブアルゴリズム

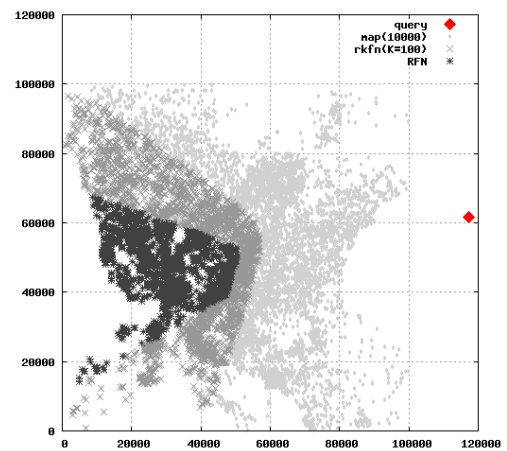


図 8 RFN と RkFN の比較 (K=100)

しているのに対し、提案手法では安定して高速に処理を行っていることがわかる。図 6 は map のデータセットの場合であるが、un や rc でも同等な結果が出た。結果として、提案手法はナイーブなアルゴリズムに比べ大幅な効率化が可能なことを実証できた。

また、RFarN の解として許容する範囲 d に応じた計算コストの変化を検証した。データサイズ 10000 のそれぞれのデータセットにおける結果を図 7 に示す。

図 7 の結果から、合成データ・実データともに許容する範囲 d が大きいほど計算コストがかかることがわかる。これは d が大きくなるほど、効率化アルゴリズムのフィルタリングで解から排除できる点が少なくなってしまうためであり、妥当な結果といえる。

4.2 RkFN

RkFN においても同様に 3.4 の提案手法とナイーブなアルゴリズムの速度と検索結果を比較した。結果を図 8, 9 に示す。

図 8 は許容する解の個数は $k = 100$ としたときの RkFN の解を RFN と比較した結果である。こちらも RFarN と同様に、尺度が調整できるため、RFN に比べ柔軟な検索であるといえる。

図 9 の結果から、提案手法はナイーブなアルゴリズムに比べ 2,3 倍程度の速度の向上がみられる。しかし、データ数の増大とともに計算コストも大きく増えているため、更なる改良の余地があるといえる。

また、許容する解の個数 k を変化させることによる計算コストを検証した。データサイズは 10000 とし、それぞれのデータセットにおける結果を図 10 に示す。

データセットの種類によってばらつきがみられるが、基本的にはこちらも RFarN と同様に、許容量を大きくするほど計算コストがかかることがわかる。

4.3 考察

以上の実験結果から、RFarN に関しては十分な速度の向上がみられている。一方で、RkFN に関してはさらなる速度の向上が見込まれる。スケーラビリティの観点からも、実世界では更に大きなデータを扱うことが考えられるため、RkFN はさらなる効率化のために今後新たなアルゴリズムの考案が検討される。

5. 結論

本研究では、類似検索の一つ、遠方検索において、従来の逆

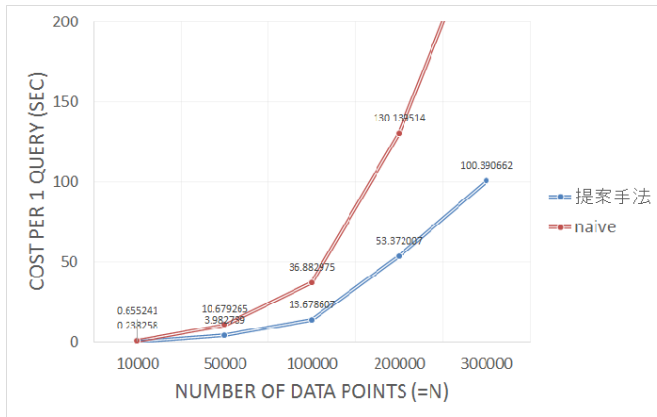


図 9 RkFN 提案手法 VS ナイーブアルゴリズム

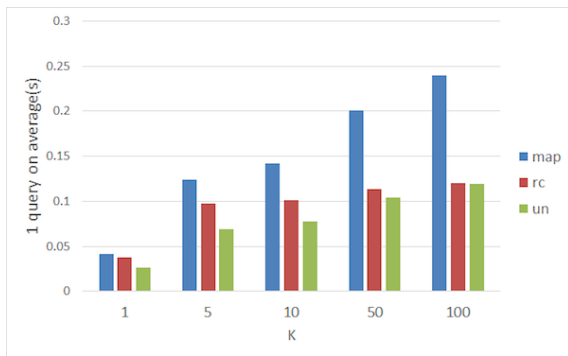


図 10 K に応じた計算コスト

最遠方検索の問題点を指摘し、新たに 2 つの逆遠方検索手法を提案した。1 つは範囲での許容 (RFarN)、もう 1 つは個数での許容 (k 逆最遠方) である。

また、それぞれの逆遠方検索手法に関する効率的なアルゴリズムを実装し、ナイーブなアルゴリズムに比べて効率的であることを示した。

今後の課題として、更なる速度の向上のための RkFN の効率的な検索アルゴリズムの改良と、本提案手法をより実用的なものとするために、I/O コストの考慮も含めた手法の改良が挙げられる。

謝 辞

本研究の一部は科学研究費補助金基盤研究 (C)(24500105) による。

文 献

- [1] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509-522, 2002.
- [2] Kristen Grauman and Trevor Darrell. Fast contour match-

- ing using approximate earth mover's distance. In *CVPR* (1), pages 220-227, 2004.
- [3] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Ze Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2), 2008.
- [4] Rakesh Agrawal, Christos Faloutsos, and Arun N. Swami. Efficient similarity search in sequence databases. In *FODO*, pages 69-84, 1993.
- [5] Bin Yao, Feifei Li, and Piyush Kumar, reverse furthest neighbors in spatial databases. In *ICDE*, pages 664-675, 2009.
- [6] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In *SIGMOD Conference*, pages 47-57, 1984.
- [7] Nick Roussopoulos, Stephen Kelley, and Fredeic Vincent. Nearest neighbor queries. In *SIGMOD Conference*, pages 71-79, 1995.
- [8] King Lum Cheung and Ada Wai-Chee Fu. Enhanced nearest neighbor search on the r-tree. *SIGMOD Record*, 27(3):16-21, 1998.
- [9] Gisli R. Hjaltason and Hanan Samet. Distance browsing in spatial databases. *ACM Trans. Database Syst.*, 24(2):265-318, 1999.
- [10] Vassilis Athitsos, Michalis Potamias, Panagiotis Papapetrou, and George Kollios. Nearest neighbor retrieval using distance-based hashing. In *ICDE*, pages 327-336, 2008.
- [11] Yufei Tao, Ke Yi, Cheng Sheng, and Panos Kalnis. Quality and efficiency in high dimensional nearest neighbor search. In *SIGMOD Conference*, pages 563-576, 2009.
- [12] Jianquan Liu, Hanxiong Chen, Kazutaka Furuse, and Hiroyuki Kitagawa. An efficient algorithm for reverse furthest neighbors query with metric index. In *DEXA*, pages 437-451, 2010.