D-008

類似検索の高速化を目的とした Pivot 選択手法の実験評価
Empirical Evaluation of Pivot Selection Methods for Fast Similarity Search

倉沢 央 [1]　深川 大路 [2]　高須 淳宏 [3]　安達 淳 [3]
Hisashi Kurasawa　Daiji Fukagawa　Atsuhiro Takasu　Jun Adachi

1. はじめに

類似検索は，膨大なオブジェクトの中からクエリに似たオブジェクトを探す技術である．類似検索は画像検索やデータベースのクリーニングなど様々なアルゴリズムの高速化に役に立つ．我々は距離空間(メトリック空間)で扱える距離(類似度)を対象とした類似検索の高速化を目指し，類似検索索引の研究開発に取り組んでいる．本索引は距離の公理を満たす類似度すべてを対象とし，ベクトル間のユークリッド距離や文字列間の編集距離などを扱える．

類似検索索引は，三角不等式などの距離の公理を利用して，オブジェクト群の中からクエリから距離の遠いオブジェクトを判別し，枝刈りするのに利用される．類似検索索引は，枝刈れるオブジェクトが多いほど検索時に発生する距離計算コストを抑えられ，早く検索できる．多くの類似検索索引で， *Pivot* と呼ばれる参照オブジェクトからの距離で，索引対象のオブジェクト集合を再帰的に部分セットに分割する索引付け手法が用いられている[1]．従来手法では， Pivot は空間の端に位置するオブジェクトを選ぶという経験則[2]やオブジェクトの分布のクラスタ構造[3]をもとに選択されていた．しかしながら，いずれの手法も特定の分布をしたオブジェクトのみに効果があり，汎用性に欠けていた．つまり，ユーザは扱うオブジェクトの分布をみて，それを得意とする索引手法を選ばねばならなかった．

そこで，我々はあらゆる分布のオブジェクトに効果的な類似検索索引， Pivot Capacity Tree (PCTree)を提案した[2]．同時に，我々は分布の特徴を調べるために， Pivot と分割距離に対する索引木のバランスに及ぼす効果と検索時の枝刈りに及ぼす効果についての評価関数， Pivot Capacity (PC)を定義した． PCTree は， PC を最大にする Pivot でオブジェクト集合を再帰的に分割し，木構造の索引を構築する．

本稿では PCTree の簡単な説明の後，索引性能についての評価実験の結果を紹介する．実験では，人工的に生成したベクトルデータの他に，先行研究の実験で用いられていた 3 つの実データを使って，検索時に発生する距離計算コストを測った．いずれのデータに対しても提案手法は高速に検索できることを示せた．

2. Pivot Capacity Tree

PCTree はあらゆる分布のオブジェクトに対して高速に類似検索を実行することを目的とした索引手法である．

2.1 Pivot Capacity

PCTree が Pivot を選択する際に参照する評価関数、PC について述べる． PC は既知のクエリ分布に対して期待できる索引性能を表す． PC はオブジェクト集合と Pivot と

1 東京大学　The University of Tokyo
2 同志社大学　Doshisha University
3 国立情報学研究所　National Institute of Informatics

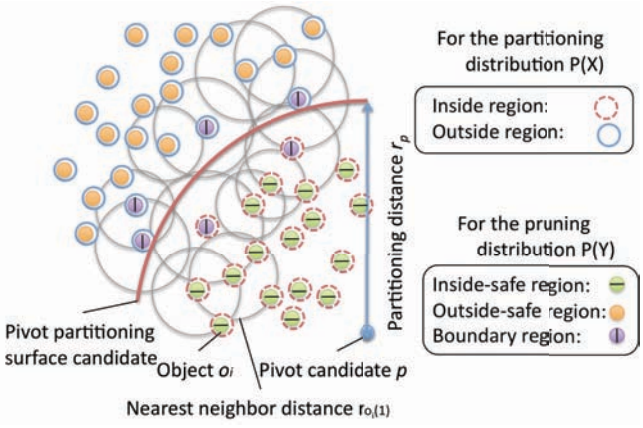

図 1　Pivot Capacity のための空間分割

の距離から 2 種類の確率変数を計算し，その相互情報量から導く．ここでは，距離空間 $M=(D,d)$において Pivot p とその分割距離 r_p が D に含まれる部分空間 R を 2 分割していると想定する．なお，本稿ではクエリの分布はオブジェクトの分布と同一のものと仮定する．

索引木のバランスについての確率変数

p と rp によって R が分割された 2 つの部分空間は，

$$R_1(p,r_p)=\{o\in R \mid d(o,p)\le r_p\},$$

$$R_2(p,r_p)=\{o\in R \mid d(o,p)> r_p\},$$

と表せる．我々は，前者を *Inside region* ，後者を *Outside region* と呼ぶ．さて，クエリ q が属する部分空間について確率変数 X で表す． q が $R_1(p,r_p)$ ， $R_2(p,r_p)$に属するときをそれぞれ $X_{1,p,rp}$， $X_{2,p,rp}$ とする． S_R を R に含まれるオブジェクト集合とすると， X についての確率は，

$$P(X_{1,p,r_p})=\frac{|S_{R_1(p,r_p)}|}{|S_R|},\quad P(X_{2,p,r_p})=\frac{|S_{R2(p,r_p)}|}{|S_R|},$$

で表せる．

枝刈りについての確率変数

オブジェクト集合 S においてオブジェクト o から k 近傍点までの距離 $r_{o,k,S}$ と定義する．これを使い， R を以下の 3 つの部分空間に分ける．

$$R'_1(p,r_p,k)=\{o\in R \mid d(o,p)+r_{o,k,S}\le r_p\},$$

$$R'_2(p,r_p,k)=\{o\in R \mid d(o,p)-r_{o,k,S}> r_p\},$$

$$R'_3(p,r_p,k)=R-R'_1(p,r_p,k)-R'_2(p,r_p,k).$$

我々はこれらを順に *Inside-safe region* ， *Outside-safe region* ， *Boundary region* と呼ぶ．これら部分空間は，検索時に枝刈れる範囲の違いを表している．具体的には，クエリ q が Inside-safe region に属するとき， k 近傍点を検索する際に Outside region に含まれるオブジェクト集合を枝刈れる．同

様に， q が Outside-safe region に属するときは Inside region に含まれるオブジェクト集合を枝刈れる． q が $R'_1(p,r_p)$ ， $R'_2(p,r_p)$ ， $R'_3(p,r_p)$のどの空間に属するかを確率変数 Y で $Y_{1,p,rp}$， $Y_{2,p,rp}$， $Y_{3,p,rp}$ で表し，その確率を求めると，

$$P(Y_{1,p,r_p}) = \frac{|S_{R'_1(p,r_p)}|}{|S_R|}, \quad P(Y_{2,p,r_p}) = \frac{|S_{R'_2(p,r_p)}|}{|S_R|},$$

$$P(Y_{3,p,r_p}) = \frac{|S_{R'_3(p,r_p)}|}{|S_R|},$$

となる．

Pivot Capacity の計算

pivot p の PC の値は， $I(;)$を相互情報量とすると，

$$PC_k(p) \equiv \max_{r_p} I(X;Y)$$

と定義した．これは，二元消失通信路の通信路容量と同様のモデルである．

2.2 索引付けと検索処理

PCTree の索引付けは，オブジェクト集合に対して PC の値を最大化するオブジェクトを Pivot に設定し， 2 つの部分集合に分割することを再帰的に実行して，木構造の索引を構築する． Pivot を選ぶ手順を除いた索引付け手法は，先行研究の VPT[2]と同じである．索引付けコストの削減を目的とした Pivot 候補のサンプリング手法については[4]を参照されたい．

検索処理では，クエリと Pivot との距離をもとに三角不等式で索引木の探索を枝刈りする．これも VPT と同じ手順である．本稿では詳細は省略する．

3. 評価実験

実験には，以下のデータセットを使用した．

- 人工ベクトル： 2 から 64 次元，クラスタ数 100 ，分散値 0.02 の混合正規分布， 100,000 件．
- NASA ： 20 次元のヒストグラムデータ， 40,150 件．
- Corel Image Datasets ： 32 次元の画像ヒストグラムデータ，68,040 件．
- ミュンヘン大の画像データセット： 112 次元のヒストグラムデータ， 112,544 件

いずれもユークリッド距離を類似度とし，データセットと同じ分布で重複のない 1,000 件をクエリとして用いた．比較手法には， Metric Space Library[5]で提供されている GHT[6]， MVP[7]， LC[8]，そして SAT[9]を使用した．各比較手法について簡単に紹介すると， GHT はオブジェクト集合の分割に 2 つの Pivot を使う手法である． MVP は VPT の改善手法で， 1 つの Pivot でオブジェクト集合を 2 つ以上の部分集合に分割する手法である． LC は Pivot によって小さな部分集合に分割するリスト構造の索引であり，SAT はグラフ構造の索引である．手法の評価尺度は[9]と同様に，検索実行時に生じる距離計算の回数を用いた．実験結果はクエリ 1,000 件の平均値である．結果の図の縦軸は Naïve な Sequential Scan を実行した際の距離計算回数を 1 としたときの比を表している．図 2 の横軸は次元数を，図 3 から 5 の横軸はクエリが要求する近傍オブジェクト数を表している．

図 2 から 5 はそれぞれのデータセットに対する結果を示している．図 2 から低次元では MVP や GHT が，高次元では LC が優れた性能を発揮していることがわかる． SAT の検索コストは次元数の増大にかかわらずほぼ変わらないが，他手法よりも大きい．これに対して， PCTree は次元数にかかわらず常に検索コストの削減に最も効果を発揮していることがわかる．

実データを用いた実験では，図 3 と 5 では MVP が，図 4 では LC の性能が索引の効果を発揮している． GHT は NASA のデータに， SAT は Corel のデータに対して性能が悪い．一方， PCTree はいずれのデータに対しても他手法を上回る性能であることがわかる．

これより，提案手法の PCTree はデータの分布にかかわらず類似検索の高速化を実現できていることが確認できた．

4. まとめ

本稿では，距離空間を対象にした類似検索索引， PCTree について述べた．人工データと 3 つの実データを用いた実験結果より， PCTree は関連研究と比較して，あらゆる分布のオブジェクトに対して検索コスト削減に効果的な索引であることを確認できた．

参考文献

[1]P.Zezula, et al. Similarity Search: The Metric Space Approach. Springer-Verlag, (2005).
[2]P.N.Yianilos. Data Structure and Algorithms for Nearest Neighbor Search in General Metric Spaces. In SODA, (1993).
[3]H.V.Jagadish, et al. iDistance: An Adaptive b+-tree based Indexing Method for Nearest Neighbor Search. ACM Trans. On Database Systems, Vol.30, No.2, (2003).
[4]H.Kurasawa, et al. Pivot Selection Method for Optimizing both Pruning and Balancing in Metric Space Indexes. In DEXA, (2010).
[5]Metric Space Library, http://www.sisap.org/Metric_Space_Library.html.
[6]J.K.Uhlman. Satisfying General Proximity / Similarity Queries with Metric Trees. Information Processing Letters, Vol.40, No.4, (1991).
[7]T.Bozkaya, et al. Indexing Large Metric Spaces for Similarity Search Queries. ACM Trans. On Database Systems, Vol.24, No.3, (1999).
[8]E.Chevez, et al. A Compact Space Decomposition for Effective Metric Indexing. Pattern Recognition Letters, Vol.24, No.9, (2005).
[9]G.Navarro. Searching in Metric Spaces by Spatial Approximation. The VLDB Journal, Vol.11, No.1, (2002).

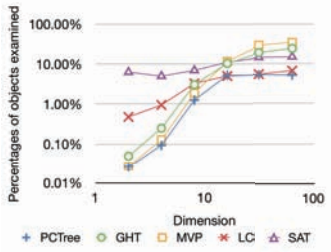

図 2　人工ベクトル

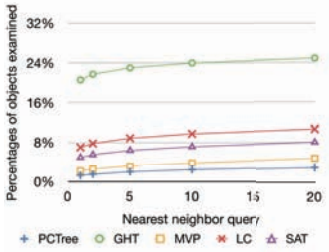

図 3　NASA

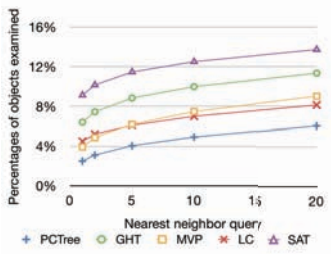

図 4　Corel Image Datasets

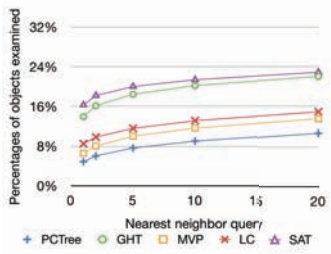

図 5　ミュンヘン大