D-027

ソートを利用したＬ１ノルム総和計算によるピボット選択の高速化
Speeding-up pivot selection based on summation of L1-distances by sorting

小林えり † 伏見卓恭 † 斉藤和巳 † 池田哲夫 †
Eri Kobayashi Takayasu Fushimi Kazumi Saito Tetsuo Ikeda

1. はじめに

近年，Web 上には多量のデータが蓄積されており，与えられたクエリから類似したオブジェクトを検索する類似検索研究の重要性は益々高まっている．類似検索とは，クエリと類似したオブジェクトをデータベースなどの中から検出する問題を指す．オブジェクト間の類似度は距離関数から求まり距離関数は，非負性，対称性，および三角不等式の性質を満たす．データの多くは高次元で表現されるが，高次元空間に存在するオブジェクト間の距離を計算するには大量の計算を行わなければならない．そのため，類似検索ではこの計算量を削減し，検索を高速化させるために一部のオブジェクトを選定して導くピボット集合を用いる方法が提案されている．本稿ではピボット選択の高速化を目的に距離ソートを利用し，最適なピボット集合を構成する手法を提案する．実データを用いて従来法と提案法を比較し，提案法の有効性を述べる．

2. 類似検索問題

類似検索にはいくつかの手法があり，本稿ではクエリから一定のレンジ内にあるオブジェクトを検出するレンジクエリ法を扱う．レンジクエリは，オブジェクト集合 $X = \{x_1 \cdots x_N\}$ とクエリ $q \in X$ とレンジ r が与えられたとき q と x_n の距離 $d(x_n, q)$ が r 以下となるようなオブジェクト集合を求める手法である．本稿では，レンジクエリ計算時間を短縮させるためにピボット法を用いた．ピボット法はオブジェクト間の距離計算回数を削減し，検索を高速化させるために一部のオブジェクトを選定してピボット集合を構築していく手法である．以下に距離計算回数を削減できる理由を記述する．

集合 X の中から k 個のピボット集合 $P = \{p_1 \cdots p_K\}$ を選定する．オブジェクト間の距離は距離公理の三角不等式により式 1，式 2 が成立する．

$$d(q, x_n) > d(q, p) - d(x_n, p) \quad (1)$$

$$d(q, x_n) > d(x_n, p) - d(q, p) \quad (2)$$

さらに距離公理の対称性条件より式 3 が各ピボット p_k で成立し，クエリとピボットとの距離 $d(q, p)$ からクエリとオブジェクトとの距離 $d(q, x_n)$ 下界値を算出することができる．

$$d(q, x_n) > |d(q, p) - d(x_n, p)| \quad (3)$$

クエリ q に対し，ピボット集合 P による最大の距離下界値を以下のように定義する．

$$D(q, x_n; P) = \max_{1 \le k \le K} |d(q, p_k) - d(x_n, p_k)| \quad (4)$$

下界値が r 以上のオブジェクト集合 L に対し距離計算は不要となるため距離計算時間の短縮が期待できる．

$$L = \{x_n : D(q, x_n; P) > r\} \quad (5)$$

Bustos らは文献 [1] より良いピボット集合の指標として目的関数を定義し，目的関数の最大化させるピボット集合を求める最適化問題として定式化した．ピボット法において，より良いピボット集合を構成するには以下の関数を最大化させるようなピボットを選定することが求められる．

$$\begin{aligned} \mathcal{F}(P) &= \sum_{n=1}^{N-1} \sum_{m=n+1}^{N} D(x_n, x_m; P) \\ &= \sum_{n=1}^{N-1} \sum_{m=n+1}^{N} \max_{1 \le k \le K} |d(x_n, p_k) - d(x_m, p_k)| \end{aligned} \quad (6)$$

†静岡県立大学

3. 提案アルゴリズム

提案法は，ピボットとオブジェクト間の距離のソートを利用し，ピボット選択を高速化する．具体的には，我々の提案した頑健射影法 [2] における，オブジェクトペア距離の L1 ノルム総和計算の高速化と類似したアイデアを利用し，オブジェクト数の自乗オーダ $O(N^2)$ の計算量を線形対数オーダ $O(N \log N)$ に削減する．本稿では，ピボット数が 2 までの範囲でのアルゴリズムについて述べる．

まず，ピボット数が 1 のとき，ピボット p_1 とオブジェクト x_n に対し，オブジェクト番号集合

$$A(x_n) = \{m \mid d(x_n, p_1) < d(x_m, p_1)\}$$

を定義すれば次式を得る．

$$\begin{aligned} &|d(x_n, p_1) - d(x_m, p_1)| \\ &= \begin{cases} d(x_m, p_1) - d(x_n, p_1) & m \in A(x_n), \\ d(x_n, p_1) - d(x_m, p_1) & m \notin A(x_n). \end{cases} \end{aligned}$$

すなわち，$d(x_n, p_1)$ はマイナス符号で $|A(x_n)|$ 回，プラス符号で $N - 1 - |A(x_n)|$ 回出現する．ここで，$|A(x_n)|$ は集合 $A(x_n)$ の要素数を表し，$|A(x_n)| + 1$ はピボット p_1 との距離でオブジェクトを降順にソートしたときの $d(x_n, p_1)$ の順位に他ならない．よって，$d(x_n, p_1)$ のマイナス符号とプラス符号での出現を相殺すれば，ピボット数が 1 のときの目的関数を次式で計算できる．

$$\begin{aligned} \mathcal{F}(\{p_1\}) &= \sum_{n=1}^{N-1} \sum_{m=n+1}^{N} |d(x_n, p_1) - d(x_m, p_1)| \\ &= \sum_{n=1}^{N} (N - 1 - 2|A(x_n)|) d(x_n, p_1) \end{aligned} \quad (7)$$

次に，ピボット数が 2 のとき，ピボット p_1 と p_2 に対して，オブジェクト番号集合

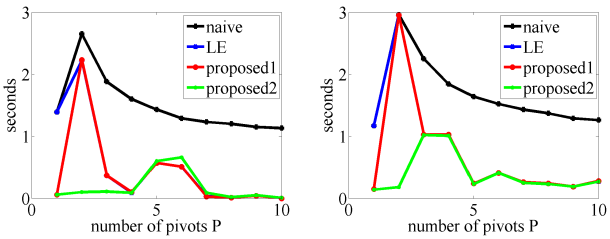
$$\begin{aligned} B(x_n) &= \{m \mid d(x_n, p_1) + d(x_n, p_2) \\ &\qquad < d(x_m, p_1) + d(x_m, p_2)\}, \\ C(x_n) &= \{m \mid d(x_n, p_1) - d(x_n, p_2) \\ &\qquad < d(x_m, p_1) - d(x_m, p_2)\}. \end{aligned}$$

を定義すれば次式を得る．

$$\begin{aligned} &\max_{p \in \{p_1, p_2\}} |d(x_n, p) - d(x_m, p)| \\ &= \begin{cases} d(x_m, p_1) - d(x_n, p_1) & m \in B(x_n) \cap C(x_n), \\ d(x_m, p_2) - d(x_n, p_2) & m \in B(x_n) \setminus C(x_n), \\ d(x_n, p_2) - d(x_m, p_2) & m \in C(x_n) \setminus B(x_n), \\ d(x_n, p_1) - d(x_m, p_1) & m \notin B(x_n) \cup C(x_n). \end{cases} \end{aligned}$$

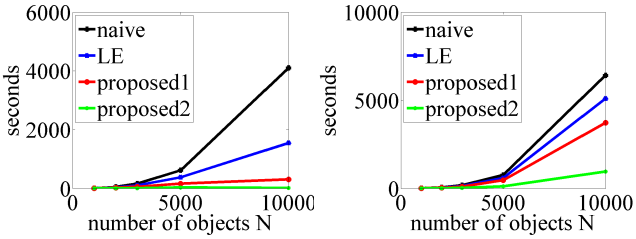
すなわち，$d(x_n, p_1)$ はマイナス符号で $|B(x_n) \cap C(x_n)|$ 回，プラス符号で $N - 1 - |B(x_n) \cup C(x_n)|$ 回出現するので，これらを相殺すれば，$d(x_n, p_1)$ に対する係数は以下となる．

$$\begin{aligned} &N - 1 - |B(x_n) \cup C(x_n)| - |B(x_n) \cap C(x_n)| \\ &= N - 1 - (|B(x_n)| + |C(x_n)|) \end{aligned}$$



(a) @cosme (b) MNIST

図 1: 各ピボット選択に有した計算時間



(a) @cosme (b) MNIST

図 2: ピボットを 3 つ選定するまでに要した累計計算時間

同様に，$d(x_n, p_2)$ はマイナス符号で $|B(x_n) \setminus C(x_n)|$ 回，プラス符号で $|C(x_n) \setminus B(x_n)|$ 回出現するので，これらを相殺すれば，$d(x_n, p_2)$ に対する係数は以下となる．

$$\begin{aligned} & |C(x_n) \setminus B(x_n)| - |B(x_n) \setminus C(x_n)| \\ = \; & |C(x_n) \setminus B(x_n)| + |B(x_n) \cap C(x_n)| \\ & -(|B(x_n) \setminus C(x_n)| + |B(x_n) \cap C(x_n)|) \\ = \; & |C(x_n)| - B(x_n)| \end{aligned}$$

したがって，ピボット数が 2 のときの目的関数を次式で計算できる．

$$\begin{aligned} \mathcal{F}(\{p_1, p_2\}) &= \sum_{n=1}^{N-1} \sum_{m=n+1}^{N} \max_{p \in \{p_1, p_2\}} |d(x_n, p) - d(x_m, p)| \\ &= \sum_{n=1}^{N} (N - 1 - (|B(x_n)| + |C(x_n)|)) d(x_n, p_1) \\ &\quad + \sum_{n=1}^{N} (|C(x_n)| - |B(x_n)|) d(x_n, p_2) \end{aligned} \tag{8}$$

ここで，$|B(x_n)| + 1$ および $|C(x_n)| + 1$ は，ピボット p_1 と p_2 との距離の和および距離の差でオブジェクトを降順にソートしたときの順位に他ならない．すなわち，式 7 や式 8 の目的関数値を単純に計算すれば，オブジェクト数 N の自乗オーダ（$O(N^2)$）の計算量が必要となる．これに対し，提案法については，$|A(x_n)|$ ，または，$|B(x_n)|$ および $|C(x_n)|$ は，オブジェクト数 N に対し計算量 $O(N \log N)$ のソートで求めることができる．そして，これらの値が求まれば，式 7 や 8 の右辺最終式は線形オーダで計算できる．よって，提案法を用いれば，オブジェクト数 N に対して $O(N \log N)$ の計算量で目的関数値を求めることができる．

4. 評価実験

4.1. 実験データ

今回の実験データとして @cosme MNIST を用いた．@cosme は美容に関する商品をアイテムとするレビューサイトである．48,548 アイテム，45,024 ユーザー，331,084 レビューを有する．MNIST は手書き文字認識用データベースのことであり，”0,1,2,3,4,5,6,7,8,9”の 10 のクラスにより表されている．”0,1,2,3,4,5,6,7,8,9”の 10digits の手書き文字の 1 つが，28 × 28=784 画素に，各画素 0〜255 の 256 階調グレースケールで表されている．

4.2. 実験設定

本稿では目的関数 6 を最大化させる手法として文献 [1] の貪欲法で求める方法を naive 法，文献 [3] の遅延評価による高速化を図った方法を LE 法とし提案法と比較する．提案法のうち，1 つ目のピボット選択にのみ提案法アルゴリズムを適応させた方法を proposed1 法，2 つ目のピボット選択まで提案法を適応させた方法を proposed2 法とする．なお，提案法はピボットが 2 つまでのときに適応されるため，提案アルゴリズムが適応されない場合のピボット選択は LE 法と同じアルゴリズムを採用している．

4.3. 評価結果

ピボットを 10 に設定した場合の各ピボット選択に要した計算時間でそれぞれの手法を評価していく．実験ではアイテム数 $N = 1,000$ とし@cosme はレビュー数の多い上位 1,000 アイテム，MNIST はランダムに 1,000 アイテム抽出したデータを用いた．横軸はピボット数を表し，縦軸はそれぞれのピボット選択に費やした時間を秒単位で表している．@cosme の実験結果 図 1(a) と MNIST の実験結果 図 1(b) を見てみると，両者とも提案アルゴリズムが適用される 1 つ目，2 つ目のピボット選択の計算時間においては提案法が naive 法，LE 法に比べて非常に短い時間で同じピボット，つまり目的関数値を最大化させるようなピボットの選定に成功している．

次にオブジェクト数を増加させていったとき，ピボットを 3 つ選択するまでに費やした累積計算時間で評価していく．こちらは横軸はオブジェクト数，縦軸は累計計算時間を秒単位で表している．@cosme ではレビュー数の多い上位から順にアイテムを抽出し，MNIST では全データからランダムにアイテムを抽出している．@cosme の実験結果 図 2(a)，MNIST の実験結果 図 2(b) を見ていく．$N = 10,000$ のときの累計計算時間を見てみると MNIST の実験結果では，proposed2 法は約 1,000 秒，naive 法は約 7,000 秒かかっており，提案法は単純法の約 1/7 の計算時間で最良なピボット集合を選定している．@cosme の実験結果では，naive 法は約 4,000 秒かかっているのに対し提案法，proposed1 法は 500 秒，proposed2 法は 100 秒もかからずに同等の結果を出力している．また両実験データともオブジェクト数を逐次増加していっても提案法が最も短い計算時間で最良なピボット集合を構成することに成功している．

5. おわりに

今回 2 つの実データを用いて提案法と従来法を比較し，提案法の有効性を述べた．今後はさらに多様なデータでの実験も行い，提案法の有効性を検証していく．

謝辞 本研究は, 科学研究費補助金基盤研究 (C)(No.23500128) の補助を受けた．

参考文献

[1] B. Bustos, G. Navarro, and E. Chavez.: ”Pivot Selection Techniques for Proximity Searching in Metric Spaces”, Proc. of Pattern Recognition Lettes, Vol.24, No.14, pp. 2357-2366, (2003)

[2] 小林 えり，伏見 卓恭，斉藤和巳，池田哲夫: ”頑健線形射影法の特性評価”，the 27th Annual Conference of the Japanese Society for Artficial Intelligence，2013，(2013)

[3] 三津山 雅規，斉藤和巳，池田哲夫，大久保誠也，武藤伸明: ”類似検索における遅延評価を用いたピボット選択の高速化”，DEIM Forum2011，Vol.9-6，(2013)