

一般化ピボット法の距離定義による特性評価

小林 えり[†] 伏見 卓恭^{††} 齊藤 和巳^{††} 池田 哲夫^{††}

[†] 静岡県立大学経営上学部 〒422-8526 静岡県静岡市駿河区谷田 52-1

^{††} 静岡県立大学経営情報イノベーション研究科 〒422-8526 静岡県静岡市駿河区谷田 52-1

E-mail: †{b10034,j11507,k-saito,t-ikeda}@u-shizuoka-ken.ac.jp

あらまし 類似検索の高速化を目的にピボット法が提案されており、従来のピボット法ではピボットを与えられたオブジェクト集合から選択していた。これに対し、ピボットをオブジェクト集合の中から選択するのではなく、オブジェクト空間から最適なピボットを構成する一般化ピボット法が提案されている。これまでの研究により一般化ピボット法はユークリッド距離定義、マンハッタン距離定義に対応可能となった。本論文ではそれぞれの距離定義による結果の違いに着目し、目的関数、計算時間、類似検索性能、可視化結果、4つの観点からそれぞれ距離定義を比較する。本実験ではマンハッタン距離定義によるものが目的関数、類似検索性能の2観点において最も優れた結果を示し、また可視化結果ではデータ間の隠れた関係性の抽出に成功したことを確認した。

キーワード 類似検索、ピボット法、マンハッタン、ユークリッド

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ますます高まっている。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められ距離関数は、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求めるのに大量の計算が必要となる。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ピボット集合として選定して利用する方法が提案されている。

効果的なピボット集合を選択する手法として Bustos らはインクリメンタル法を提案している [1]。このインクリメンタル法では、より良いピボット集合の指標として目的関数を定義し、目的関数を最大化するようにピボットを逐次追加することでピボット集合を得る。その後も、データの分布を pivot の選択に利用する iDistance やその応用、またクラスタリングが意味をなさないようなデータに対する pivot 分割の手法など多様な手法が提案されている [2], [3], [4], [5]。ただし、これらは、与えられたオブジェクト集合の中から、その部分集合としてピボット群を選択する手法である。

これに対し、オブジェクト空間の任意の点をピボットとして求める一般化ピボット計算法も提案されている [6]。一般に、オブジェクト集合の中からピボット集合を選択する場合と比較し、オブジェクト空間の任意の点としてピボット集合を求めれば目的関数値の向上が自然に期待できる。これまでの研究により、一般化ピボット法はユークリッド距離、マンハッタン距離の両方の距離定義に対応できるようになった [6] [7]。本論文ではユークリッド距離定義、マンハッタン距離定義による結果の違いに着目し、それぞれを評価していく。

2. 類似検索問題

類似検索には様々な手法が提案されているが、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問題を扱う。レンジクエリ問題は、オブジェクト集合 $X = \{x_1, \dots, x_N\}$ とクエリ q 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問題である。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る。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求める手法である。代表的な Bustos らの提案したインクリメンタル法では、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 $D(q, x_n; P)$ を次式で定義する。

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

ここで、ピボット集合の k 番目の要素を $p_k \in P$ 、集合の要素数を $K = |P|$ とし、 $d(\cdot, \cdot)$ は上述した元空間での距離関数を表す。以下に、式 (1) の距離下界値を用いる理由について補足する。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 (2)、式 (3) が成立する。

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

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

さらに距離公理の対称性条件より次式が各ピボット 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。

$$d(q, x_n) > |d(q, p) - d(x_n, p)| \quad (4)$$

式 (4) ではピボット数 1 だが、これを K 個に拡張すれば式 (1) が導ける。すなわち、最大下界値 $D(q, x_n; P)$ が r 以上のオブ

ジェクト集合 L を以下で定義する.

$$L = \{\mathbf{x}_n \mid D(\mathbf{q}, \mathbf{x}_n; P) > r\} \quad (5)$$

明らかに, L に属すオブジェクト集合に対しては距離計算が不要となるため, 類似検索計算時間の短縮が期待できる.

3. 一般化ピボット計算法

Bustos ら [1] は, より良いピボット集合の指標として目的関数を定義し, 目的関数を最大化させるピボット集合を求める問題として定式化した. Bustos らの手法では, 与えられたオブジェクト集合の中から目的関数を最大化するようなピボットを逐次選択する ($\mathbf{p} \in X$) [1].

$$\mathcal{F}(P) = \sum_{n=1}^{N-1} \sum_{m=n+1}^N D(\mathbf{x}_n, \mathbf{x}_m; P) \quad (6)$$

これに対し, 一般化ピボット計算法では, オブジェクト空間, ここでは H 次元のユークリッド空間 (R^H) の任意の点として, ピボット集合を構築する手法である ($\mathbf{u} \in R^H$) [6]. すなわち, 一般化ピボット計算法では, 次式の $\mathcal{F}(U)$ を最大にするピボット集合を $U = \{\mathbf{u}_1, \dots, \mathbf{u}_K\}$ を求める. オブジェクト, および, ピボットは, H 次元ベクトルで表す.

$$\mathcal{F}(U) = \sum_{n=1}^{N-1} \sum_{m=n+1}^N \max_{1 \leq k \leq K} |d(\mathbf{x}_n, \mathbf{u}_k) - d(\mathbf{x}_m, \mathbf{u}_k)| \quad (7)$$

すなわち, $U = \arg \max_U \mathcal{F}(U)$ を求める問題として定式化される.

いま, 任意のオブジェクトのペア $\mathbf{x}_n$ と $\mathbf{x}_m$ に対し, ピボット $\mathbf{u}_k$ で距離の下界値が最大化されるオブジェクトの番号のペアからなる集合を以下で定義する.

$$S_k(U) = \{\{n, m\} \mid k = \arg \max_{1 \leq \kappa \leq K} |d(\mathbf{x}_n, \mathbf{u}_\kappa) - d(\mathbf{x}_m, \mathbf{u}_\kappa)|\} \quad (8)$$

さらに, オブジェクト番号 n に着目したとき, $S_k(U)$ において, n とペアで出現するオブジェクト番号の集合 $S_{k,n}(U)$ を次のように定義する.

$$S_{k,n}(U) = \{m \mid \{n, m\} \in S_k(U)\} \quad (9)$$

このとき, $|d(\mathbf{x}_n, \mathbf{u}_k) - d(\mathbf{x}_m, \mathbf{u}_k)|$ の絶対値を距離の大小で外すとき, 距離 $d(\mathbf{x}_n, \mathbf{u}_k)$ がプラス符号で現れる回数と, マイナス符号での回数を相殺した係数 $c_{n,k}(U)$ は以下となる.

$$c_{n,k}(U) = |\{m \in S_{k,n}(U) \mid d(\mathbf{x}_n, \mathbf{u}_k) > d(\mathbf{x}_m, \mathbf{u}_k)\}| - |\{m \in S_{k,n}(U) \mid d(\mathbf{x}_n, \mathbf{u}_k) < d(\mathbf{x}_m, \mathbf{u}_k)\}| \quad (10)$$

ここで, ピボット候補集合を $\bar{U}(\bar{U} = \{\bar{\mathbf{u}}_1 \dots \bar{\mathbf{u}}_K\})$ とし, 式 10 を用いて計算した目的関数は以下ようになる.

$$\mathcal{F}(U|\bar{U}) = \sum_{k=1}^K \sum_{n=1}^N c_{n,k}(\bar{U}) d(\mathbf{x}_n, \mathbf{u}_k) \quad (11)$$

詳しくは各距離定義ごとの手法説明にて記述するが, 一般化ピボット法はアルゴリズムを反復することで最適なピボット集合

を構築する. 式 11 はピボット候補集合 $\bar{U}$ で得られた $c_{n,k}(\bar{U})$ を用いて目的関数を最大化させるピボット集合を求めている. $c_{n,k}(U)$ はピボット集合が更新される度に变化するため, ピボット集合が更新されれば $c_{n,k}(U)$ もまた更新される. 反復によって求めた, 目的関数値を向上させる更新ピボット集合を $\hat{U}$ とすると以下の関係が成立する.

$$\mathcal{F}(\bar{U}) = \mathcal{F}(\bar{U}|\bar{U}) \leq \mathcal{F}(\hat{U}|\bar{U}) \leq \mathcal{F}(\hat{U}|\hat{U}) = \mathcal{F}(\hat{U}) \quad (12)$$

関数 $\mathcal{F}(\hat{U})$ は更新後のピボット集合 $\hat{U}$ を用いて計算した係数 $c_{n,k}(\hat{U})$ での目的関数である. 明らかに更新ピボット集合での係数 $c_{n,k}(\hat{U})$ を用いたほうが目的関数の向上する.

3.1 ユークリッド距離定義に基づく一般化ピボット計算法

ユークリッド距離に基づくオブジェクト $\mathbf{x}_n$ と $\mathbf{x}_m$ の距離は次式で与えられることを確認する.

$$d(\mathbf{x}_n, \mathbf{x}_m) = \sqrt{\sum_{h=1}^H (x_{n,h} - x_{m,h})^2} \quad (13)$$

ここで, $x_{n,h}$ はオブジェクト $\mathbf{x}_n$ の第 h 次元の値を表す. 各ピボット $\mathbf{u}_k$ ($k = 1, \dots, K$) を逐次代入法 (Successive Substitution Method) による更新で求めることが出来る. つまり, 目的関数 $\mathcal{F}(U|\bar{U})$ を各ピボット $\mathbf{u}_k$ で微分し, 極値を求めることで目的関数値を改善するピボット集合 U を求める.

$$\frac{\partial \mathcal{F}(U|\bar{U})}{\partial \mathbf{u}_k} = \sum_{n=1}^N \frac{c_{k,n}(\bar{U})}{d(\mathbf{x}_n, \mathbf{u}_k)} (\mathbf{x}_n - \mathbf{u}_k) = 0 \quad (14)$$

式 14 より, ピボット $\mathbf{u}_k$ は以下の式で求めることが出来る.

$$\hat{\mathbf{u}}_k = \frac{\sum_{n=1}^N \frac{c_{k,n}(\bar{U})}{d(\mathbf{x}_n, \mathbf{u}_k)} \mathbf{x}_n}{\sum_{n=1}^N \frac{c_{k,n}(\bar{U})}{d(\mathbf{x}_n, \mathbf{u}_k)}} \quad (15)$$

各ピボットは式 15 で更新させていく. 以下にユークリッド距離に基づく一般化ピボット法のアルゴリズムを示す.

- L2PVL1 初期化: ピボット集合 U をランダムに構築する.
- L2PVL2 距離計算: 任意の n と k のペアに対して $d(\mathbf{x}_n, \mathbf{u}_k)$ を式 (13) で計算する.
- L2PVL3 係数計算: 各オブジェクトの係数 $c_{n,k}(\bar{U})$ を式 (10) で計算する.
- L2PVL4 ピボット更新: 各ピボット u_k ($k = 1, \dots, K$) を式 (15) により更新する.
- L2PVL5 判定: $\frac{\mathcal{F}(\hat{U}) - \mathcal{F}(\bar{U})}{\mathcal{F}(\bar{U})} \leq \varepsilon$ となるか, 反復数が最大反復回数を超えれば終了, さもないければ, ステップ L2PVL4 へ戻る.

本実験では $\varepsilon = 10^{-8}$ とし, 最大反復回数は 1000 とした.

3.2 コサイン類似度を用いた L2 一般化ピボット計算法

本論文では比較対象として, オブジェクト間の距離にコサイン類似度を用いたユークリッド距離に基づく一般化ピボット法を加える. コサイン類似度を距離に用いた場合, オブジェクト $\mathbf{x}_n$ と $\mathbf{x}_m$ の距離は次式で与えられることを確認する.

$$d(\mathbf{x}_n, \mathbf{x}_m) = \sqrt{1 - \sum_{h=1}^H \frac{x_{n,h}}{\|\mathbf{x}_n\|} \frac{x_{m,h}}{\|\mathbf{x}_m\|}} \quad (16)$$

オブジェクト間の距離をコサイン類似度としているため、関数式 11 には $\frac{1}{2} \sum_{k=1}^K \lambda_k (1 - \mathbf{u}_k^T \mathbf{u}_k)$ の制約条件が付く。

$$\tilde{\mathcal{F}}(U|\bar{U}) = \mathcal{F}(U|\bar{U}) + \frac{1}{2} \sum_{k=1}^K \lambda_k (1 - \mathbf{u}_k^T \mathbf{u}_k) \quad (17)$$

ユークリッド距離に基づく一般化ピボット法同様、逐次代入法にて目的関数値を改善するピボット集合 U を求める。目的関数 $\mathcal{F}(U|\bar{U})$ を各ピボット $\mathbf{u}_k$ で微分すると以下の式となる。

$$\frac{\partial \tilde{\mathcal{F}}(U|\bar{U})}{\partial \mathbf{u}_k} = - \sum_{n=1}^N \frac{c_{k,n}(\bar{U})}{d(\mathbf{x}_n, \mathbf{u}_k) \|\mathbf{x}_n\|} \mathbf{x}_n - \lambda_k \mathbf{u}_k = 0 \quad (18)$$

ピボット $\mathbf{u}_k$ は式 18 より以下の式で更新される。

$$\mathbf{u}_k = \frac{\sum_{n=1}^N \frac{c_{k,n}(\bar{U})}{d(\mathbf{x}_n, \mathbf{u}_k)} \frac{\mathbf{x}_n}{\|\mathbf{x}_n\|}}{\left\| \sum_{n=1}^N \frac{c_{k,n}(\bar{U})}{d(\mathbf{x}_n, \mathbf{u}_k)} \frac{\mathbf{x}_n}{\|\mathbf{x}_n\|} \right\|} \quad (19)$$

目的関数の最適化は式 (19) を用いてユークリッド距離でのものと同様のアルゴリズムで求める。

3.3 マンハッタン距離定義に基づく一般化ピボット計算法

マンハッタン距離に基づくオブジェクト $\mathbf{x}_n$ と $\mathbf{x}_m$ の距離は次式で与えられることを確認する。

$$d(\mathbf{x}_n, \mathbf{x}_m) = \sum_{h=1}^H |x_{n,h} - x_{m,h}| \quad (20)$$

ここで、 $x_{n,h}$ はオブジェクト $\mathbf{x}_n$ の第 h 次元の値を表す。また、マンハッタン距離定義で最適化する目的関数の表現は、ユークリッド距離定義によるもの [6] と同様に式 (7) となることに注意されたい。ここで式 (11) は以下のように書き換えることが出来る。

$$\begin{aligned} \mathcal{F}(U|\bar{U}) &= \sum_{k=1}^K \sum_{n=1}^N c_{n,k}(\bar{U}) d(\mathbf{x}_n, \mathbf{u}_k) \\ &= \sum_{k=1}^K \sum_{h=1}^H \sum_{n=1}^N c_{n,k}(\bar{U}) |x_{n,h} - u_{k,h}| \\ &= \sum_{k=1}^K \sum_{h=1}^H \mathcal{F}(u_{k,h}|\bar{U}) \end{aligned} \quad (21)$$

すなわち、 $\mathcal{F}(u_{k,h}|\bar{U})$ は次式で定義される。

$$\mathcal{F}(u_{k,h}|\bar{U}) = \sum_{n=1}^N c_{n,k}(\bar{U}) |x_{n,h} - u_{k,h}| \quad (22)$$

式 (7) で定義した目的関数の最大化は、各 $u_{k,h}$ に対し、独立に式 22 を最大化できることが分かる。

以下には、式 (22) の最大化法について述べる。いま、オブジェクト番号は $x_{1,h} \leq \dots \leq x_{N,h}$ となるよう再設定したとする。また、便宜上、 $x_{0,h} = -\infty$, $x_{N+1,h} = \infty$ とする。このとき、 $m = 0, \dots, N$ とし、 $x_{m,h} < u_{k,h} < x_{m+1,h}$ に対して、式 (22) は次のように変形できる。

$$\mathcal{F}(u_{k,h}|\bar{U}) = \sum_{n=1}^m c_{n,k}(\bar{U}) (u_{k,h} - x_{n,h})$$

$$\begin{aligned} &+ \sum_{n=m+1}^N c_{n,k}(\bar{U}) (x_{n,h} - u_{k,h}) \\ &= u_{k,h} \left(\sum_{n=1}^m c_{n,k}(\bar{U}) - \sum_{n=m+1}^N c_{n,k}(\bar{U}) \right) \\ &\quad - \sum_{n=1}^m c_{n,k}(\bar{U}) x_{n,h} + \sum_{n=m+1}^N c_{n,k}(\bar{U}) x_{n,h} \end{aligned} \quad (23)$$

ただし、 $n = 1$ から $m = 0$ までの和や、 $n = N + 1$ から N までの和については、何も加算されず 0 を意味するものとする。式 (23) より、目的関数 $\mathcal{F}(u_{k,h}|\bar{U})$ は区分線形となることが分かる。ここで、係数 $c_{n,k}(\bar{U})$ は $d(\mathbf{x}_n, \mathbf{u}_k)$ が加算される回数から減算される回数を引いたものであり、全オブジェクトでの総和は 0 になる。

$$\sum_{n=1}^N c_{n,k}(\bar{U}) = 0 \quad (24)$$

よって、 $u_{k,h} < x_{1,h}$ ならば、目的関数 $\mathcal{F}(u_{k,h}|\bar{U})$ は次式の一定値となる。

$$\mathcal{F}(u_{k,h}|\bar{U}) = \sum_{n=1}^N c_{n,k}(\bar{U}) x_{n,h} \quad (25)$$

また、 $x_{N,h} < u_{k,h}$ ならば、次式となる。

$$\mathcal{F}(u_{k,h}|\bar{U}) = - \sum_{n=1}^N c_{n,k}(\bar{U}) x_{n,h} \quad (26)$$

したがって、目的関数 $\mathcal{F}(u_{k,h}|\bar{U})$ の最大値は、区分線形式の交点 $\{x_{1,h}, \dots, x_{N,h}\}$ のどれかで与えられる。すなわち、ピボット k の第 h 座標値の更新式は以下となる。

$$\hat{u}_{k,h} = \arg \max_{\{x_{n,h} \mid 1 \leq n \leq N\}} \mathcal{F}(x_{n,h}|\bar{U}) \quad (27)$$

明らかに、更新後のピボット集合を $\hat{U}$ で表せば、 $\mathcal{F}(\bar{U}) \leq \mathcal{F}(\hat{U})$ の関係が成り立つ。一方、上述しているように、各ピボットの各次元での値の取り得る値の範囲は、有限離散点に限定される。よって、提案法は、常に有限回の反復で収束することが保証される。

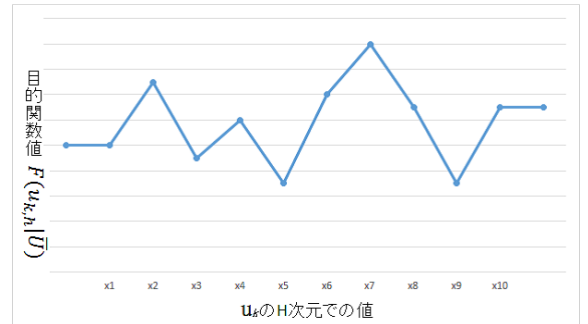


図 1 目的関数 $\mathcal{F}(\mathbf{u}_{k,h})$

図 1 には、目的関数 $\mathcal{F}(u_{k,h}|\bar{U})$ のイメージを示す。すなわち、縦軸には目的関数値、横軸に $\mathbf{u}_{k,h}$ のとりうる値を設定した

グラフである．図 1 の示すように目的関数 $\mathcal{F}(u_{k,h})$ は，各オブジェクトの第 h 座標値において傾きを変える．この図では， $x_{7,h}$ で目的関数 $\mathcal{F}(u_{k,h}|\bar{U})$ は最大化される．以下に，L1PVL 法と呼び提案アルゴリズムを示す．

L1PVL1 初期化：ピボット集合 U をランダムに構築する．

L1PVL2 ソート：各次元 h ごとにオブジェクトの座標値 $x_{n,h}$ を昇順にソートする．

L1PVL3 距離計算：任意の n と k のペアに対して $d(\mathbf{x}_n, \mathbf{u}_k)$ を式 (20) で計算する．

L1PVL4 係数計算：各オブジェクトの係数 $c_{n,k}(\bar{U})$ を式 (10) で計算する．

L1PVL5 ピボット更新：任意の k と h のペアに対して $u_{k,h}$ を式 (27) で計算する．

L1PVL6 判定：L2PVL 法と同様の終了条件を満たせば終了，さもなければ，ステップ PVL3 へ戻る．

ここで，式 (27) のピボット更新により，目的関数 (11) の値を改善させることができるとともに，ピボットの更新値は有限集合に限定される．よって，上記アルゴリズムは有限反復回数で終了することが保証される．すなわち，上記アルゴリズムの反復で目的関数 (11) を最大化するピボット集合 U を構成できる．

4. 実験による評価

4.1 実験データ

実験データとして MNIST と 毎日新聞 を用いた．目的関数，類似検索性能，可視化結果で用いたオブジェクトはランダムに 10,000 オブジェクト抽出したものを使用している．

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．これらの 10digits の手書き文字の 1 つが， $28 \times 28=784$ 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．すなわち，各文字は 784 次元のベクトルで表現される．

毎日新聞 は日本の新聞会社のひとつであり，本実験では，1992 年から 2002 年の国際関係の記事のみを実験対象としている．オブジェクトを単語頻度ベクトルとし，次元数は記事内に出てきたターム数，51,030 とする．例えばオブジェクト x_n の記事内に”国”という単語が 2 回出現した場合，”国”に対応する次元 (仮に $h=1$ を”国”に対応する次元とする) の要素を 2，つまり $x_{n,1}=2$ とし，オブジェクトベクトル x_n を決定していく．

以下では，これらデータを用いて目的関数，計算時間，可視化結果，類似検索性能 4 つの観点からマンハッタン距離定義に基づく一般化ピボット法 (以下 L1PVL 法)，ユークリッド距離定義に基づく一般化ピボット法 (以下 L2PVL 法)，コサイン類似度を用いたユークリッド距離定義に基づく一般化ピボット法 (以下 cL2PVL 法) の 3 手法を比較する．

4.2 目的関数の比較

図 2 に目的関数での結果を示す．横軸はピボット数を，縦軸は目的関数値を表している．赤線で L1PVL 法を，青線で L2PVL 法，緑線で cL2PVL 法の結果を示す．ただし，一般にマンハッタン距離定義による目的関数値よりユークリッド距離定義による目的関数値の方が大きくなるため，そのまま比較す

るのは妥当ではない．そのためそれぞれの手法の理論値を分母に用いて，評価関数を以下のように定義した．

$$g(U) = \frac{\mathcal{F}(U)}{\sum_{n=1}^{N-1} \sum_{m=n+1}^N d(\mathbf{x}_n, \mathbf{x}_m)} \times 100 \quad (28)$$

すなわち，一般化ピボット法による距離下限値の良さをパーセントで評価する． $g(U)$ が高いほど元空間でのデータ量を保持した結果であると言える．

図 2(a) に示した MNIST の結果から，L1PVL 法がどの手法よりも高い目的関数値を算出しており，最も元空間での情報を保持した結果であることがわかった．L2PVL 法と cL2PVL 法の結果を比較してみると，ピボット構成の際，制約のかかる cL2PVL 法よりも L2PVL 法の方が高い目的関数値を算出しているのが確認できる．しかしながら L1PVL 法と比較するとどのピボット数のときでも 10 % 以上下回る結果であった．

図 2(b) に示した毎日新聞の結果から，こちらも L1PVL 法が最も高い目的関数値を算出し，L2PVL 法と cL2PVL 法の結果を比較しても，cL2PVL 法よりも L2PVL 法の方が高い目的関数値を算出しているのが確認できた．毎日新聞の結果の場合，L1PVL 法の結果が他の手法に比べて圧倒的に高い目的関数値を示し，ピボット数が $K=10$ の場合，約 30 % 近く他の手法を上回っていることが確認できる．

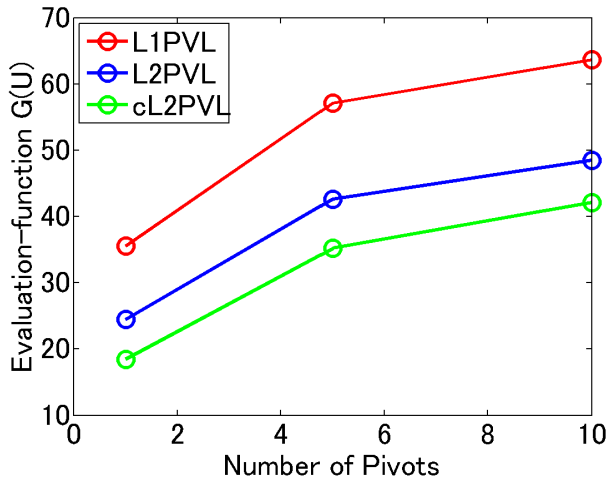
両データとも cL2PVL 法はもっとも関数値の低い結果となったが，これは cL2PVL 法はピボットを一定の範囲内の中から構成しなければならないために，制約のない，つまりは無限空間の中からピボットを構築する L1PVL 法，L2PVL 法と比較して低い目的関数値となってしまったのだと考えられる．

4.3 計算時間の比較

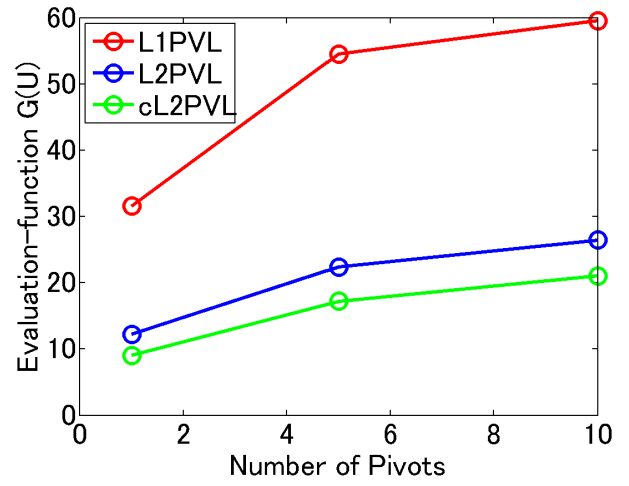
図 3 に，計算時間での結果を示す．横軸はオブジェクト数を表し，縦軸が計算時間を秒単位で表わしている．図 2 と同様に，赤線で L1PVL 法を，青線で L2PVL 法，緑線で cL2PVL 法を表すとともに， K はピボット数を表し，今回はピボット数が 1, 10 のときの結果を評価する．

図 3(a) に示した MNIST の結果から，L2PVL 法，cL2PVL 法，L1PVL 法の順に短い計算時間で結果を算出していることが分かる．L2PVL 法はピボット数に関わらず，他の手法と比べて非常に時間がかかっているのが分かる．各手法のアルゴリズム反復終了条件は，更新によって向上する目的関数値が収束するか，反復回数が上限 (本実験では 1000 回) を上回る，の 2 つである．L1PVL 法，cL2PVL 法は前者の条件で終了しているが，L2PVL 法は後者の条件で終了している．そのため他の手法でのアルゴリズム反復回数は 50～100 回に対し，L2PVL 法は上限の 1000 回であり，ゆえに他の手法と比べて圧倒的に時間がかかってしまったのだと考えられる．

図 3(b) に示した毎日新聞の結果より，L2PVL 法，L1PVL 法，cL2PVL 法の順に短い計算時間で結果を算出していることが分かる．こちらのデータセットでも L2PVL 法の結果は他の手法よりも非常に時間がかかっており，他 2 手法に対して反復回数が多いことが原因であると考えられる．L1PVL 法は一概にどの手法よりも高速に結果を算出できる，というわけではなく次元数が多いデータセットの場合だと多少時間がかかってし

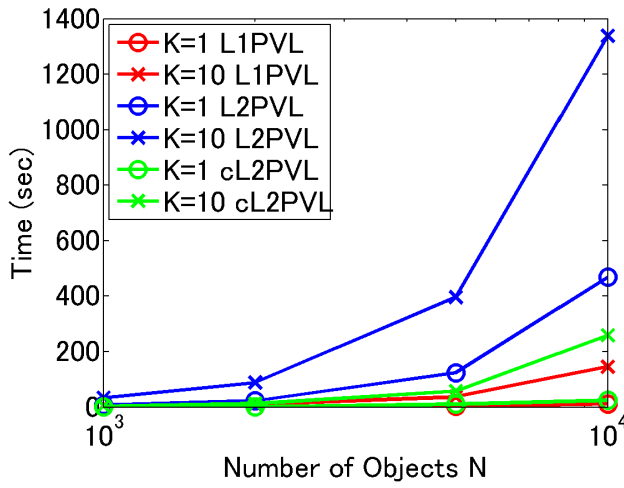


(a) MNIST

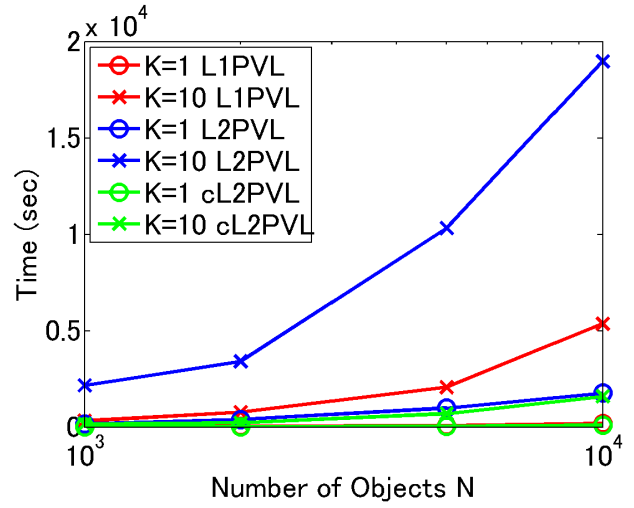


(b) 毎日新聞

図 2 目的関数

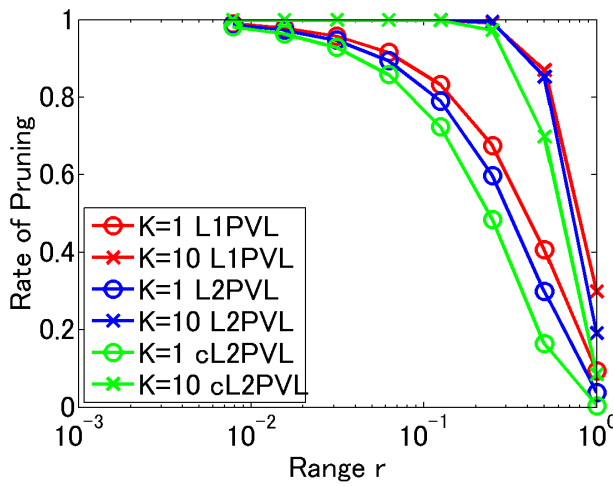


(a) MNIST

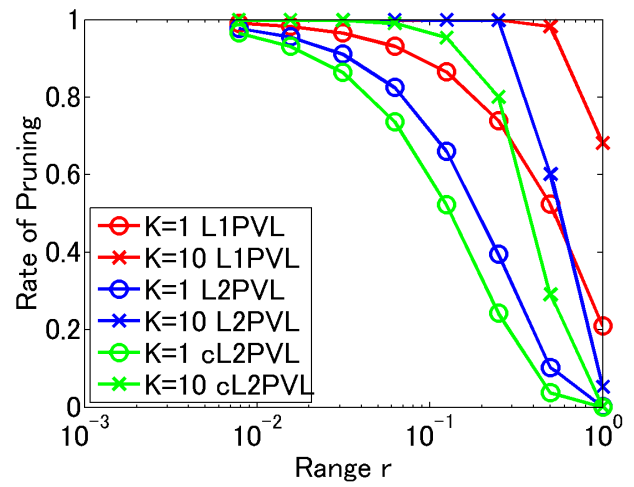


(b) 毎日新聞

図 3 計算時間

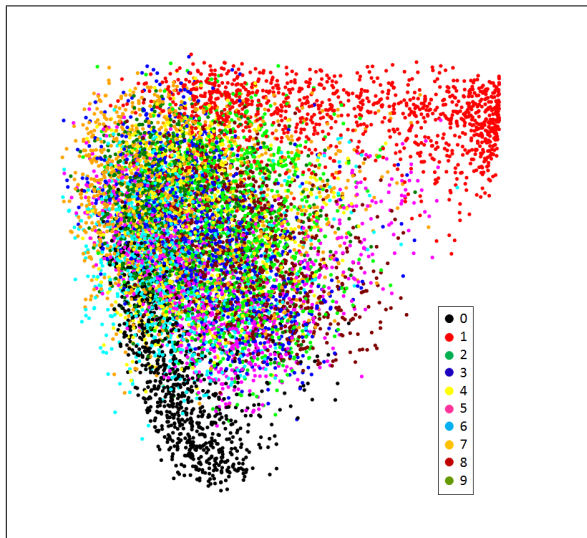


(a) MNIST

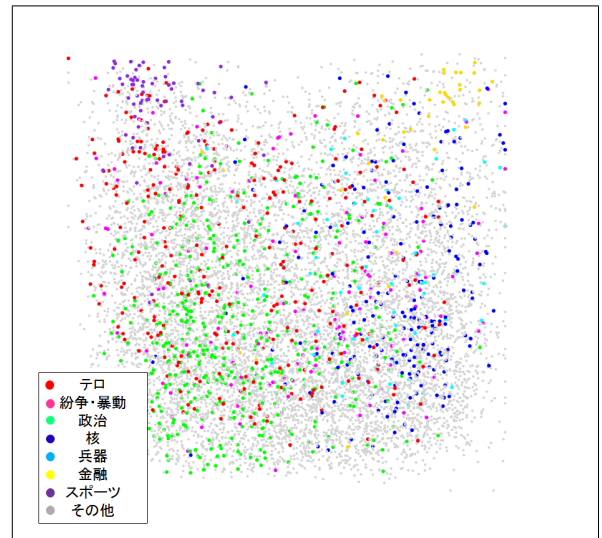


(b) 毎日新聞

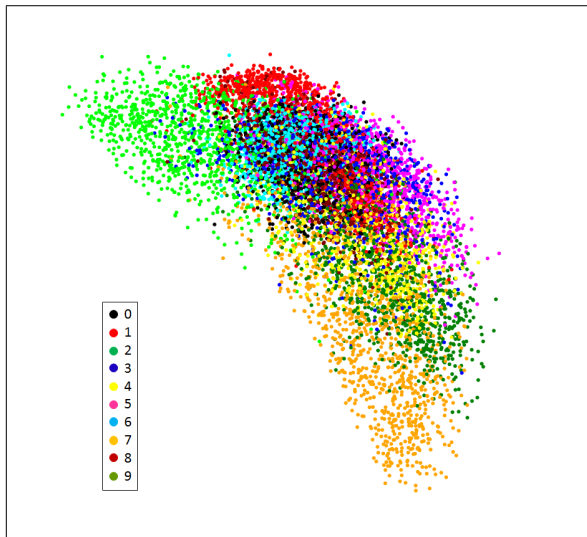
図 4 レンジクエリ



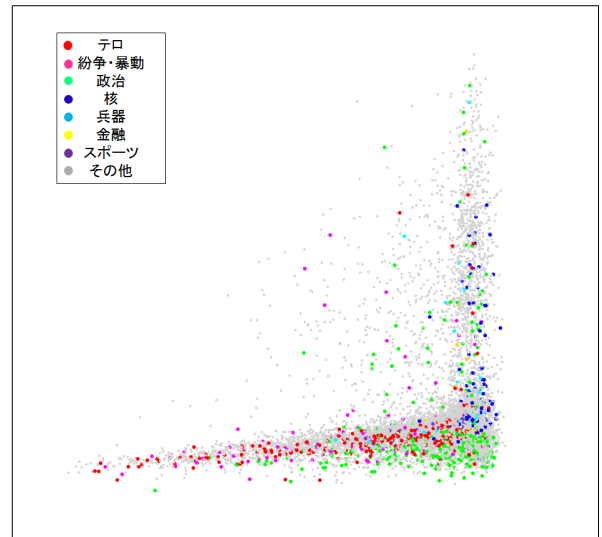
(a) L1PVL 法



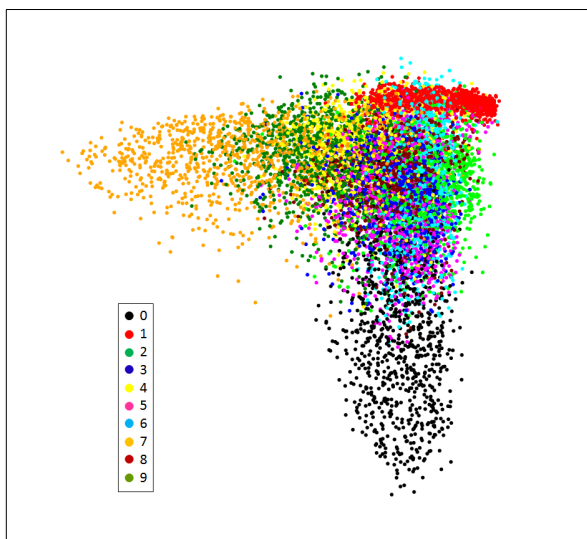
(a) L1PVL 法



(b) L2PVL 法

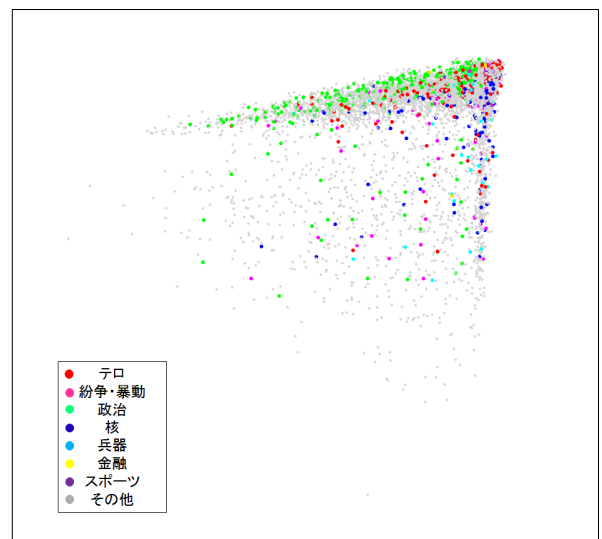


(b) L2PVL 法



(c) cL2PVL 法

図 5 MNIST での可視化結果



(c) cL2PVL 法

図 6 毎日新聞での可視化結果

まうことが分かった。

4.4 類似検索性能の比較

図 4 に、レンジクエリ問題における類似検索性能の観点で 3 手法を比較した結果を示す。横軸はレンジ r を表し、縦軸にはレンジ r としたとき、式 (5) で定義した集合 L を求め、枝刈り成功率 $|L|/N$ を表している。すなわち、縦軸の値が高ければ高いほど効果的な枝刈りが実現できることになる。これまでと同様に、赤線で L1PVL 法を、青線で L2PVL 法、緑線で cL2PVL 法を表しており、 K はピボット数を表し、今回はピボット数が 1, 10 のときの結果を評価する。

図 2(a) に示した MNIST の結果を見てみると、 $K = 1$ の場合、最も目的関数値が向上した L1PVL 法はどの手法よりも高い枝刈り成功率を算出していることが分かる。 $K = 10$ の結果の場合、レンジ距離 $r = 1.0$ 近くでは L1PVL 法はわずかに L2PVL 法より高い枝刈り成功率を示している。また cL2PVL 法はもっとも目的関数値が小さかったため、効率的な枝刈りを期待できるとはいえない結果となった。

図 3(b) に示した毎日新聞の結果では、MNIST よりもピボット数、レンジ距離に関わらず各手法間に明確な差が生じている。ピボット数に関わらず最も高い目的関数値を算出した L1PVL 法が最もデータの散らばった結果であると定量的に評価できる。よって、L1PVL 法は 3 手法の中で最も効果的な枝刈りを期待できる。L2PVL 法と cL2PVL 法の結果を比較してみると、より目的関数値が向上している L2PVL 法の結果の方が効率的な枝刈りが見込めることが分かった。

4.5 可視化による評価

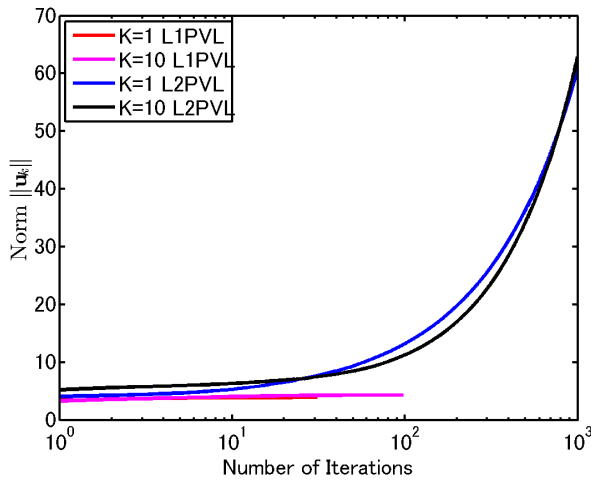
図 5 に示した MNIST の結果については、オブジェクトは "0,1,2,3,4,5,6,7,8,9" の 10 のクラスにより色分けされている。図 5(a) から結果を見てみると L1PVL 法がもっともデータが全体的に散らばった可視化結果であり、効果的な枝刈りが期待できると分かる。L1PVL 法はクラス "1" と "0" のオブジェクトが他のクラスと重ならずにまとまってプロットされている。"1" や "0" はどはっきりとは分別できないが比較的同じクラス同士は散らばらずに近傍にプロットされている。図 5(b) から L2PVL1 法の結果を見てみるとクラス "1", "2", "7" が他のクラスと被らずにまとまってプロットされているのが分かる。また形の似ている "7", "9", "4" や "0", "6" などは近傍にプロットされている。図 5(c) から cL2PVL 法の結果を見てみるとクラス "7", "0" が他のクラスとは被らずにまとまってプロットされているのが分かる。L2PVL 法と同様、同一クラス同士、形の似ているクラス同士は比較的近傍にプロットされているのが確認できた。

次に図 6 に示した毎日新聞の結果を評価する。オブジェクトは記事のジャンルによって色分けされており、収集期間の時期は北朝鮮のテポドン発射、アメリカの同時多発テロ問題、イラク戦争などの紛争問題、などといった、"北朝鮮関連", "核問題", "テロ・紛争問題" が多い時期であったことを考慮されたい。図 6(a) から L1PVL 法の結果を見てみると他の手法と比べてデータが全体的に満遍なく広がった可視化結果となった。"スポーツ", "金融", "兵器" と "核" といったジャンルはま

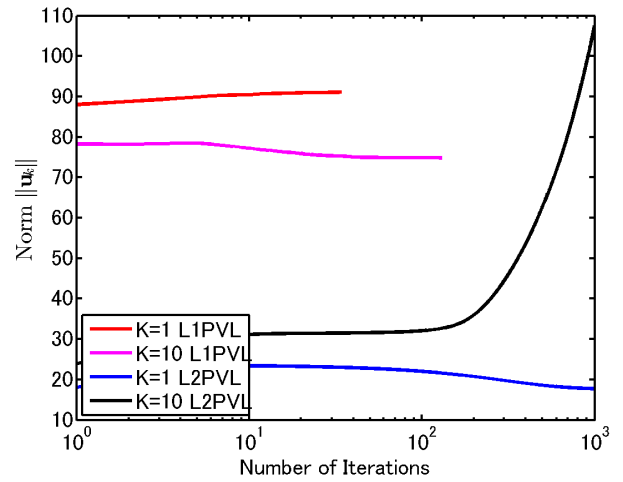
とまって近傍にプロットされているのが分かり、他の手法と比べてジャンルごとにまとまってプロットされた結果だと確認できた。また選挙など"政治"関連の記事は左下にプロットされており、"テロ"や"紛争問題"は全体に広がっているのが確認できる。下にプロットされた記事は総単語数が多く、そのため文字数の多い、特集記事や論評などが多かった。逆に、結果だけを記事とする株価変動やスポーツの記事は総単語数が少ないために上側にプロットされている。図 6(b) から L2PVL 法の結果を見てみると下側に"テロ", "紛争問題"などが抽出され、右端には"中国", "台湾", "北朝鮮"の 3 か国に関する記事が抽出された可視化結果となった。収集期間ではテロ・紛争問題や中国・台湾・北朝鮮関連の記事が多数記述されたために、その他の記事間との距離が広がり、それ故にそれ以外の記事は密集した結果となったと考えられる。図 6(c) から cL2PVL 法の結果を見てみると、上側には政治関連の記事、詳しくは"大統領"のワードが入った記事がまとまって抽出され、右端にはロシア関連の記事が多く見られた。収集期間の時期は 1991 年にソビエト連邦が崩壊しロシア連邦が成立する、などロシア関連の記事が多い時期でもあったため他の記事との距離が広がり抽出されやすかったのではないかと考えられる。

5. 考察

本実験にて、L2PVL 法は L1PVL 法よりも圧倒的に時間がかかっているのに目的関数値は L1PVL 法を大きく下回った結果となっている。図 7 はアルゴリズム反復ごとのピボット $\mathbf{u}_k$ のノルム $\|\mathbf{u}_k\|$ を表し、図 8 はアルゴリズム反復ごとの目的関数値を表している。図 7, 図 8 から L2PVL 法は最適なピボットを発見出来ていないのではないかと考えられる。L1PVL 法は早い段階で最適ピボットに近づくことが出来たため、図 7 において L1PVL 法の $\|\mathbf{u}_k\|$ はどのピボット数でもすぐに水平なグラフとなり、反復回数も少ない量で済んでいる。L1PVL 法はアルゴリズムから反復回数が有限であることが保証されているため、局所改善によって最適ピボットを求める L2PVL 法よりも最適ピボットの発見が容易である。これに対し、L2PVL 法は図 7 からピボットのノルム $\|\mathbf{u}_k\|$ は反復によって常に変化し続けており、最大反復回数を迎えてもいまだピボットは更新し続けているため、最適なピボットを見つけていないのが分かる。つまり L2PVL 法は最適なピボットを発見出来なかったために目的関数が伸び悩んだのだと判断できる。そのため最大反復回数を増やせば目的関数の向上が期待できるかもしれないが、図 3 の示す通り計算時間が膨大な量になってしまうため、実用的ではないと考えられる。図 7, 8 には載せていないが cL2PVL 法に関して、こちらも L2PVL 法同様、局所改善によって最適ピボット集合を求める手法であるが、こちらはピボット構成の際、無限空間からではなく、制約された有限空間の中から構成されるため、L2PVL 法よりも最適ピボット見つけやすい環境にある。しかしながら有限空間でのピボット構成のために目的関数値が伸び悩むという欠点の確認された。

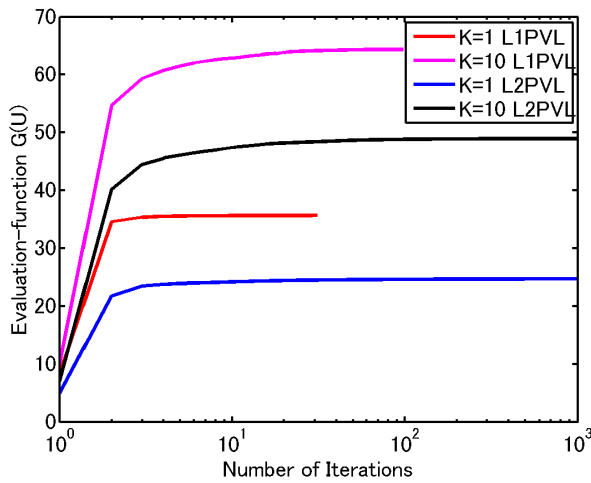


(a) MNIST

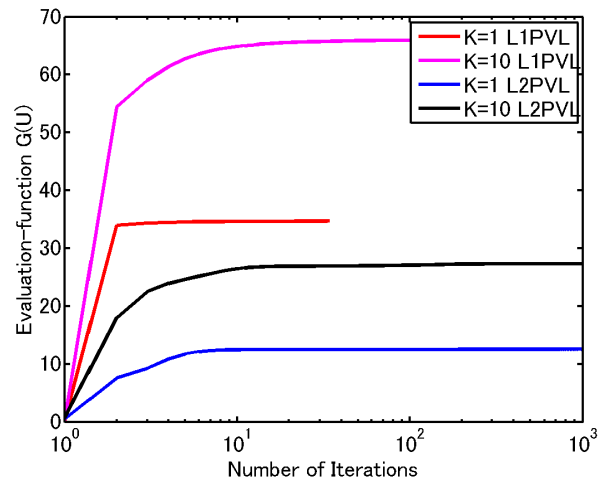


(b) 毎日新聞

図 7 反復による $\|p_k\|$ の更新



(a) MNIST



(b) 毎日新聞

図 8 反復による目的関数値の更新

6. おわりに

本論文ではマンハッタン距離定義，ユークリッド距離に基づく一般化ピボット法の結果の違いに着目し，目的関数，計算時間，類似検索性能，可視化結果，以上の4つの観点からそれぞれを比較した．マンハッタン距離定義に基づく一般化ピボット法はユークリッド距離のものよりも高い目的関数値を算出しており，効果的な枝刈りの期待できることが分かった．また，データ間の隠れた関係性を発見することのできる可視化結果を確認でき，今回の実験ではマンハッタン距離定義に基づく手法の有効性を発見することが出来た．今後は，距離定義を混合したようなデータに対応する，混合型一般化ピボット法の提案を行う．

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．

文 献

- [1] B. Bustos, G. Navarro, and E. Chavez.: "Pivot Selection Techniques for Proximity Searching in Metric Spaces",

- Pattern Recognition Lettes, Vol.24, No.14, pp. 2357-2366 (2003)
- [2] E. Chevez and G. Navarro, A compact space decomposition for effective metric indexing, Pattern Recognition Letters, 24(9), pp.1363-1376, 2005 (2005)
- [3] H. V. Jagadish, B. C. Ooi, K. L. Tran, C. Yu and R. Zhang, idistance: An adaptive b+-tree based indexing method for nearest neighbor search, ACM TODS, 30(2), pp.364-397, 2003 (2003)
- [4] H. Kurasawa, D. Fukagawa, A. Takasu and J. Adachi, Pivot selection method for optimizing both pruning and balancing in metric space indexes, Proc. DEXA, 2010 (2010)
- [5] Y. Zhuang, Y. Zhuang, Q. Li, L. Chen and Y. Yu, Indexing high-dimensional data in dual distance spaces: a symmetrical encoding approach, Proc. EDBT, 2008 (2008)
- [6] 木村 学, 斉藤和巳, 池田哲夫, 上田修功: "効率的な類似検索のためのピボット学習法", 情報処理学会論文誌, Vol.50, No.8, 1883-1891 (2009)
- [7] 小林 えり, 伏見 卓恭, 斉藤和巳, 池田哲夫: "マンハッタン距離に基づく一般化ピボット計算法", "第 6 回 Web とデータベースに関するフォーラム (WebDB2013)", Nov.2013. (2013)